



## UNIVERSIDAD TÉCNICA DEL NORTE

### FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

#### CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

#### TEMA

“APLICACIÓN WEB PARA LA ADMINISTRACIÓN ONLINE DE CITAS MÉDICAS EN EL CENTRO MÉDICO DE ORIENTACIÓN Y PLANIFICACIÓN FAMILIAR CEMOPLAF-OTAVALO; UTILIZANDO EL PATRÓN DE ARQUITECTURA MVC EN PHP”

#### APLICATIVO

“CEMOCITE SISTEMA DE CITAS MEDICAS ONLINE

CEMOPLAF- OTAVALO”

**Autor:** Amparo Alejandra Franco Nicolalde

**Director:** Ing. Miguel Orquera

Ibarra – Ecuador

2013

# *Certificación*

Certifico que la Tesis “**APLICACIÓN WEB PARA LA ADMINISTRACIÓN ONLINE DE CITAS MÉDICAS EN EL CENTRO MÉDICO DE ORIENTACIÓN Y PLANIFICACIÓN FAMILIAR CEMOPLAF-OTAVALO; UTILIZANDO EL PATRÓN DE ARQUITECTURA MVC EN PHP**” con el aplicativo “**CEMOCITE SISTEMA DE CITAS MEDICAS ONLINE CEMOPLAF- OTAVALO.**” ha sido realizada en su totalidad por la señora: Amparo Alejandra Franco Nicolalde portadora de la cédula de identidad número: 100295106-7



---

**Ing. Miguel Orquera  
Director de la Tesis**

# *Certificación*

Otavalo, 13 de diciembre del 2012

Señores

UNIVERSIDAD TÉCNICA DEL NORTE

Presente

De mis consideraciones.-

Siendo auspiciantes del proyecto de tesis de la Egresada AMPARO ALEJANDRA FRANCO NICOLALDE con CI: 1002951067 quien desarrolló su trabajo con el tema “APLICACIÓN WEB PARA LA ADMINISTRACIÓN ONLINE DE CITAS MÉDICAS EN EL CENTRO MÉDICO DE ORIENTACIÓN Y PLANIFICACIÓN FAMILIAR CEMOPLAF-OTAVALO; UTILIZANDO EL PATRÓN DE ARQUITECTURA MVC EN PHP” con el aplicativo "CEMOCITE SISTEMA DE CITAS MEDICAS ONLINE CEMOPLAF- OTAVALO ", me es grato informar que se han superado con satisfacción las pruebas técnicas y la revisión de cumplimiento de los requerimientos funcionales, por lo que se recibe el proyecto como culminado y realizado por parte de la egresada AMPARO ALEJANDRA FRANCO NICOLALDE. Una vez que hemos recibido la capacitación y documentación respectiva, nos comprometemos a continuar utilizando el mencionado aplicativo en beneficio de nuestra institución.

La egresada AMPARO ALEJANDRA FRANCO NICOLALDE puede hacer uso de este documento para los fines pertinentes en la Universidad Técnica del Norte.

Atentamente,



Dr. Lilian Cervantes

---

Dr. Lilian Cervantes  
Directora  
CEMOPLAF - OTAVALO



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**CESIÓN DE DERECHOS DE AUTOR DEL**  
**TRABAJO DE INVESTIGACIÓN**  
**A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL**  
**NORTE**

Yo, AMPARO ALEJANDRA FRANCO NICOLALDE, con cedula de identidad Nro. 1002951067, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la ley de propiedad intelectual del Ecuador, articulo 4, 5 y 6, en calidad de autor del trabajo de grado denominado: "**“APLICACIÓN WEB PARA LA ADMINISTRACIÓN ONLINE DE CITAS MÉDICAS EN EL CENTRO MÉDICO DE ORIENTACIÓN Y PLANIFICACIÓN FAMILIAR CEMOPLAF-OTAVALO; UTILIZANDO EL PATRÓN DE ARQUITECTURA MVC EN PHP”**" con el aplicativo "**“CEMOCITE SISTEMA DE CITAS MEDICAS ONLINE CEMOPLAF- OTAVALO”**", que ha sido desarrollada para optar por el título de Ingeniería en Sistemas Computacionales, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes mencionada, aclarando que el trabajo aquí descrito es de mi autoría y que no ha sido previamente presentado para ningún grado o calificación profesional.

En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica del Norte

.....  
Firma

Nombre: AMPARO ALEJANDRA FRANCO NICOLALDE

Cédula: 1002951067

Ibarra a los 13 días del mes de febrero del 2013

UNIVERSIDAD TÉCNICA DEL NORTE



BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN  
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

**1. IDENTIFICACIÓN DE LA OBRA**

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital institucional determina la necesidad de disponer los textos completos de forma digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual ponemos a disposición la siguiente investigación:

DATOS DE CONTACTO	
CEDULA DE IDENTIDAD	1002951067
APELLIDOS Y NOMBRES	AMPARO ALEJANDRA FRANCO NICOLALDE
DIRECCIÓN	Ibarra, Río Yasuní 1-119
EMAIL	<a href="mailto:ale29jun@hotmail.com">ale29jun@hotmail.com</a>
TELÉFONO FIJO	2952-750
TELÉFONO MOVIL	0985120007

DATOS DE LA OBRA	
TITULO	"APLICACIÓN WEB PARA LA ADMINISTRACIÓN ONLINE DE CITAS MÉDICAS EN EL CENTRO MÉDICO DE ORIENTACIÓN Y PLANIFICACIÓN FAMILIAR CEMOPLAF-OTAVALO; UTILIZANDO EL PATRÓN DE ARQUITECTURA MVC EN PHP"
AUTOR	AMPARO ALEJANDRA FRANCO NICOLALDE
FECHA	13 DE FEBRERO DEL 2013
PROGRAMA	PREGRADO
TITULO POR EL QUE	INGENIERÍA EN SISTEMAS COMPUTACIONALES
DIRECTOR	INGENIERO MIGUEL ORQUERA

## **2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD**

Yo, AMPARO ALEJANDRA FRANCO NICOLALDE, con cedula de identidad Nro. 1002951067, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y el uso del archivo digital en la biblioteca de la universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión, en concordancia con la Ley de Educación Superior Artículo143.



.....  
Firma

Nombre: AMPARO ALEJANDRA FRANCO NICOLALDE

Cédula: 1002951067

Ibarra a los 13 días del mes de diciembre del 2012

## *Dedicatoria*

A mi hija *Lisseth Alejandra*, quien es la luz de mis ojos, y por quien día tras día, doy mi mejor esfuerzo; a ella va dedicado con mucho amor y cariño el presente proyecto y todos los triunfos que alcance en mi vida, de igual manera a mi amado esposo que es el pilar de apoyo para cada logro alcanzado.

*Alejandra.*

## *Agradecimiento*

Agradezco a Dios quien ha guiado mis pasos entre fracasos y éxitos, dándome la fuerza para superar cada tropiezo, entregándome la enseñanza del valor, la lucha y la perseverancia.

A mi esposo y a mi hija, quienes me han apoyado en todas mis metas propuestas y han sido el pilar para alcanzar mis objetivos.

A mis padres que me han dado la vida y la fuerza para soñar en metas alcanzables.

A mi director de Tesis Ing. Miguel Orquera por su apoyo en cada paso de la elaboración de la tesis, además de su confianza y apoyo en este proyecto.

A mis amigos, de la Universidad que durante tantos años nos ha unido un fuerte lazo de amistad, forjado en las aulas de clase, de manera especial a Lucia Oñate por estar siempre a mi lado apoyándome e incentivándome para alcázar mis metas demostrándome que la amistad verdadera es para siempre.

A la Universidad Técnica del Norte y en especial a la Facultad de Ingeniería en Ciencias Aplicadas y a la Carrera de Ingeniería en Sistemas Computacionales por permitirme ser parte de una generación de triunfadores y gente productiva para el país.

Finalmente a todas aquellas personas que me ayudaron de manera desinteresada, por su apoyo, tiempo e información para el logro de mis objetivos.

¡Gracias!...

# INDICE DE CONTENIDOS

Certificación	II
Certificación	III
CESIÓN DE DERECHOS	IV
AUTORIZACIÓN DE USO Y PUBLICACIÓN	V
Dedicatoria	VII
Agradecimiento	VIII
INDICE DE CONTENIDOS	IX
INDICE DE TABLAS	XI
INDICE DE FIGURAS	XII
RESUMEN	14
SUMMARY	15
CAPÍTULO 1	17
Antecedentes, situación actual y prospectiva	17
1.1 INTRODUCCIÓN	18
1.2 ANTECEDENTES	18
1.3 SITUACIÓN ACTUAL	19
1.4 DEFINICIÓN DEL PROBLEMA	20
1.5 PROSPECTIVA DEL PROBLEMA	21
1.6 OBJETIVOS DEL PROYECTO	21
Objetivo General	21
Objetivos Específicos	21
1.7 JUSTIFICACIÓN DEL PROYECTO	22
1.7.1 Justificación Tecnológica	22
1.7.2 Justificación Metodológica	24
1.8 ALCANCE Y LIMITACIONES DEL PROYECTO	24
1.8.1 Módulos Y Relaciones	25
CAPÍTULO 2	27
Marco teórico	27
2.1 METODOLOGIA DE DESARROLLO	28
2.1.1 INTRODUCCION	28
2.1.2 RUP (Proceso Unificado Rational)	29
2.1.3 Características RUP	29
2.1.4 ESTRUCTURA DINÁMICA DEL PROCESO FASES E INTERACCIONES	30
2.1.5 ESTRUCTURA ESTÁTICA DEL PROCESO, ROLES, ACTIVIDADES, ARTEFACTOS Y FLUJOS DE TRABAJO	33
2.1.6 INTEGRACION DE FASES Y ACTIVIDADES EN RUP	35
2.2 PHP (Hypertext Preprocessor)	36
2.2.1 Aspectos a considerar	37
2.2.2 Arquitectura y Servicios	38
2.3 FRAMEWORKS DE DESARROLLO WEB	38
2.3.1 INTRODUCCION	38
2.3.2 Framework PHP Codelgniter	39
2.3.3 Historia y Evolución Codelgniter	40
2.3.4 Funcionalidades de Codelgniter	40
2.3.5 Flujo de la Aplicación de Codelgniter	40
2.4 PATRONES DE DISEÑO WEB	41
2.4.1 INTRODUCCIÓN	41
2.4.2 Patrón de Arquitectura MVC	42
2.4.3 Diseño y Arquitectura	42
2.5 GESTORES DE BASES DE DATOS	44
2.5.1 Administración de Recursos	44
2.5.2 Aspectos a considerar	45
2.5.3 Arquitectura y Servicios	46
2.5.4 Mysql	46

2.5.5	Servicios Móviles (SMS Gateway)	47
2.6	HERRAMIENTAS DE DESARROLLO	48
2.6.1	STARUML 5.0.2	48
2.6.2	Power Designer 11.0	49
2.6.3	SQL Front	50
2.6.4	Zend Studio 9.0	51
2.6.5	Librería Php Envío SMS	52
CAPÍTULO 3		55
	Desarrollo del sistema	55
3.1	INTRODUCCIÓN	56
3.2	VISIÓN	57
3.2.1	PROPOSITO	57
3.2.2	ALCANCE	58
3.2.3	OPORTUNIDAD DE NEGOCIO	58
3.3	MODELO DE NEGOCIOS	59
3.4	ESPECIFICACIONES DE REQUERIMIENTOS	60
3.4.1	Requerimientos de Entorno	61
3.4.2	Requerimientos funcionales	61
3.4.3	Requerimientos no funcionales	63
3.4.4	Descripción de actores	63
3.4.5	Descripción de casos de uso	65
3.5	ANALISIS Y DISEÑO	75
3.5.1	Modelo de análisis	75
3.5.2	Arquitectura del sistema	88
3.5.3	Diseño de casos de uso	88
3.5.4	Diagrama de clases de diseño	95
3.5.5	Modelo de datos	96
3.6	IMPLEMENTACION	98
3.6.1	Definición de subsistemas de implementación	99
3.6.2	Desarrollo de la aplicación	99
3.6.3	Implementación del sitio web	100
3.7	PRUEBAS AL SISTEMA	100
3.7.1	Casos de prueba	101
3.7.2	Evaluación de los resultados de pruebas	106
3.8	DESPLIEGUE	108
3.8.1	Plan de Despliegue	109
3.8.2	Paso a producción	109
3.8.3	Transferencia de tecnología y capacitación	111
CAPÍTULO 4		112
	CONCLUSIONES Y RECOMENDACIONES	112
4.1	CONCLUSIONES	113
4.2	RECOMENDACIONES	115
	Glosario de términos y bibliografía	117
	Glosario de términos	118
	Bibliografía	121
	anexos	123
	ANEXOS	124

## **INDICE DE TABLAS**

Tabla 2.1. Flujos de Trabajo	35
Tabla 3.1. Sentencias que definen el problema	59
Tabla 3.2. Sentencias que definen la posición del Producto	59
Tabla 3.3. Descripción de actores	64
Tabla 3.4. Caso de Uso Módulo Administración: Autenticar usuario Administrador	66
Tabla 3.5. Caso de Uso Módulo Administración: Cambiar password administrador.	66
Tabla 3.6. Caso de Uso Módulo Administración: Registrar Especialidad médica	66
Tabla 3.7. Caso de Uso Módulo Administración: Listar especialidad médica.	67
Tabla 3.8. Caso de Uso Módulo Administración: Modificar especialidad médica	67
Tabla 3.9. Caso de Uso Módulo Administración: Registrar usuarios	68
Tabla 3.10. Caso de Uso Módulo Administración: Modificar usuarios	68
Tabla 3.11. Caso de Uso Módulo Administración: Listar usuarios	68
Tabla 3.12. Caso de Uso Módulo Administración: Ver usuarios	68
Tabla 3.13. Caso de Uso Módulo Administración: Inactivar usuarios	69
Tabla 3.14. Caso de Uso Módulo usuarios: Autentificar paciente.	69
Tabla 3.15. Caso de Uso Módulo usuarios: Cambiar password paciente.	69
Tabla 3.16. Caso de Uso Módulo usuarios: Reservación de cita médica	70
Tabla 3.17. Caso de Uso Módulo usuarios: Ver reservación de citas médicas	70
Tabla 3.18. Caso de Uso Módulo usuarios: Cancelar reservación de cita médica	71
Tabla 3.19. Caso de Uso Módulo usuarios: Autentificar profesional médico.	72
Tabla 3.20. Caso de Uso Módulo usuarios: Cambiar password médico.	72
Tabla 3.21. Caso de Uso Módulo usuarios: Listar reservaciones que deben ser atendidas por el médico.	72
Tabla 3.22. Caso de Uso Módulo usuarios: cita atendida por el profesional médico.	73
Tabla 3.23. Caso de Uso Módulo usuarios: Autentificar secretaria	74
Tabla 3.24. Caso de Uso Módulo usuarios: Cambiar password médico.	74
Tabla 3.25. Caso de Uso Módulo usuarios : Registrar, modificar, listar y ver pacientes	74
Tabla 3.26. Caso de Uso Módulo usuarios: Reservar cita médica solicitada por el paciente	75
Tabla 3.27. Identificación de paquetes de análisis	87
Tabla 3.28. Formato de descripción de casos de prueba	101
Tabla 3.29. Caso de prueba: Autentificar usuario administrador	101
Tabla 3.30. Caso de prueba: Cambiar password del administrador	102
Tabla 3.31. Caso de prueba: Registrar especialidad médica	102
Tabla 3.32. Caso de prueba: Listar especialidad médica	102
Tabla 3.33. Caso de prueba: Cambiar especialidad médica	103
Tabla 3.34. Caso de prueba: Ingreso de usuarios estándar	103
Tabla 3.35. Caso de prueba: Autentificar usuario paciente	103
Tabla 3.36. Caso de prueba: Reservar cita médica por especialidad y médico	104
Tabla 3.37. Caso de prueba: Cancelar reservación por el usuario paciente	104
Tabla 3.38. Caso de prueba: Autentificar médico	104
Tabla 3.39. Caso de prueba: Ver reservaciones médicas asignadas al profesional médico	105
Tabla 3.40. Caso de prueba: Autentificar secretaría	105
Tabla 3.41. Caso de prueba: Ver reservaciones de citas médicas	105
Tabla 3.42. Caso de prueba: Reservar cita solicitada por el paciente	106
Tabla 3.43. Caso de prueba: Eliminar reservaciones de citas médicas	106
Tabla 3.44. Diseño de la tabla evaluación de resultados de prueba	107
Tabla 3.45. Evaluación de resultados de pruebas para el usuario administrador	107
Tabla 3.46. Evaluación de resultados de pruebas para el paciente	108
Tabla 3.47. Evaluación de resultados de pruebas para el médico	108
Tabla 3.48. Evaluación de resultados de pruebas para la secretaria	108
Tabla 3.49. Actividades de paso a producción	110

## **INDICE DE FIGURAS**

Figura 1.1. Misión, Visión y Valores CEMOPLAF	19
Figura 1.2. Modelo Vista Controlador (MVC)	23
Figura 1.3. Arquitectura Planteada	24
Figura 1.4. Diagrama de Módulos y Relaciones	26
Figura 2.1. Fases e hitos en RUP	30
Figura 2.2. Relación entre roles, actividades, artefactos	34
Figura 2.3. Esfuerzo en actividades según fase del proyecto	35
Figura 2.4. Solicitud de una página dinámica a través de su URI	38
Figura 2.5. Flujo de datos a través del sistema	41
Figura 2.6. Arquitectura MVC	42
Figura 2.7. Servicios móviles Gateway SMS.	48
Figura 2.8. StarUML 5.0.2	49
Figura 2.9. Power Designer 11.0	50
Figura 2.10. SQL-Front	51
Figura 2.11. Zend Studio 9.0	52
Figura 2.12. Esquema Físico Eclipsoft	53
Figura 2.13. Librería PHP para el envío de SMS	54
Figura 3.1. Diagrama del Modelo de Negocios CEMOCITE	60
Figura 3.2. Caso de uso Autentificar usuario administrador	65
Figura 3.3. Caso de uso Gestionar Especialidad Médica	66
Figura 3.4. Caso de uso Gestionar usuarios (paciente, médico secretaria)	67
Figura 3.5. Caso de uso Autentificar paciente.	69
Figura 3.6. Caso de uso Reservación de citas médicas.	70
Figura 3.7 Caso de uso Cancelar reservación de citas médicas	71
Figura 3.8. Caso de uso Autentificar profesional médico	71
Figura 3.9. Caso de uso Listar reservaciones que deben ser atendidas por el médico.	72
Figura 3.10. Caso de uso cita atendida por el profesional médico.	73
Figura 3.11. Caso de uso Autentificar secretaria	73
Figura 3.12. Caso de uso Ingresar paciente	74
Figura 3.13. Caso de uso reservar cita médica solicitada por el paciente	75
Figura 3.14. Diagrama de secuencia del CU: Autentificar usuario administrador	76
Figura 3.15. Diagrama de secuencia del CU: cambiar password usuario administrador	76
Figura 3.16. Diagrama de secuencia del CU: Registrar especialidad médica	77
Figura 3.17. Diagrama de secuencia del CU: Cambiar estado de la especialidad médica	77
Figura 3.18. Diagrama de secuencia del CU: Listar especialidad médica	78
Figura 3.19. Diagrama de secuencia del CU: Registrar Usuario estándar	78
Figura 3.20. Diagrama de secuencia del CU: Modificar Usuario estándar	79
Figura 3.21. Diagrama de secuencia del CU: Listar Usuario estándar	79
Figura 3.22. Diagrama de secuencia del CU: Eliminar Usuario estándar	80
Figura 3.23. Diagrama de secuencia del CU: Ver Usuario estándar	80
Figura 3.24. Diagrama de secuencia del CU: Autentificar paciente	81
Figura 3.25. Diagrama de secuencia del CU: Reservar cita médica	81
Figura 3.26. Diagrama de secuencia del CU: Ver Reservación de cita médica	82
Figura 3.27. Diagrama de secuencia del CU: Cancelar cita médica	82
Figura 3.28. Diagrama de secuencia del CU: Autentificar médico	83
Figura 3.29. Diagrama de secuencia del CU: Listar reservaciones que deben ser atendidas por el profesional médico	83
Figura 3.30. Diagrama de secuencia del CU: Cita atendida por el profesional médico	84
Figura 3.31. Diagrama de secuencia del CU: Autentificar secretaria	84
Figura 3.32. Diagrama de secuencia del CU: Ingresar paciente	85
Figura 3.33. Diagrama de secuencia del CU: Reservar cita médica solicitada por el paciente	85
Figura 3.34. Diagrama de secuencia del CU: Cancelar cita médica	86
Figura 3.35. Dependencia entre paquetes de análisis	88
Figura 3.36. Arquitectura Planteada	88
Figura 3.37. Diagrama de colaboración del CU: Autentificar usuario administrador	89
Figura 3.38. Diagrama de colaboración del CU: cambiar password usuario administrador	89
Figura 3.39. Diagrama de colaboración del CU: Registrar especialidad médica	89
Figura 3.40. Diagrama de colaboración del CU: Eliminar especialidad médica	90

---

Figura 3.41. Diagrama de colaboración del CU: Listar especialidad médica	90
Figura 3.42. Diagrama de colaboración del CU: Registrar Usuario estándar	90
Figura 3.43. Diagrama de colaboración del CU: Modificar Usuario estándar	90
Figura 3.44. Diagrama de colaboración del CU: Listar Usuario estándar	91
Figura 3.45. Diagrama de colaboración del CU: Eliminar Usuario estándar	91
Figura 3.46. Diagrama de colaboración del CU: Ver Usuario estándar	91
Figura 3.47. Diagrama de colaboración del CU: Autenticar paciente	92
Figura 3.48. Diagrama de colaboración del CU: Reservar cita médica	92
Figura 3.49. Diagrama de colaboración del CU: Ver Reservación de cita médica	92
Figura 3.50. Diagrama de colaboración del CU: Cancelar Reservación de cita médica	93
Figura 3.51. Diagrama de colaboración del CU: Autenticar médico	93
Figura 3.52. Diagrama de colaboración del CU: Listar reservaciones que deben ser atendidas por el médico	93
Figura 3.53. Diagrama de colaboración del CU: Cita atendida por el profesional médico	94
Figura 3.54. Diagrama de colaboración del CU: Autenticar secretaria	94
Figura 3.55. Diagrama de colaboración del CU: Ingresar paciente	94
Figura 3.56. Diagrama de colaboración del CU: Reservar cita médica solicitada por el paciente	95
Figura 3.57. Diagrama de colaboración del CU: Cancelar Reservación de cita médica	95
Figura 3.58. Diagrama de clases del sistema CEMOCITE	96
Figura 3.59. Modelo lógico de la base de datos	97
Figura 3.60. Modelo físico de la base de datos	98
Figura 3.61. Modelo de Implementación	99
Figura 3.62. Pantalla de ingreso al sistema	100
Figura 3.63. Diagrama d Despliegue para el sistema CEMOCITE	109

## **RESUMEN**

Esta tesis titulada “Aplicación web para la Administración Online de Citas Médicas en el Centro Médico de Orientación y Planificación Familiar Cemoplaf-Otavalo; Utilizando el Patrón de Arquitectura MVC en php”, optimiza y estandariza el flujo de la información dentro de cada uno de los procesos de manejo de usuarios, reservación de citas médicas, administración de médicos, secretarías, creación de especialidades médicas. Fue desarrollado utilizando herramientas libres y de alta tecnología como son Codeigniter, MySQL, Php JavaScript y AJAX, además de incorporar una completa y sofisticada metodología de desarrollo de software denominada: RUP ((Rational Unified Process ).

El objetivo del Sistema de administración de citas online es contar con una herramienta informática eficiente que permita la obtención de información verídica y actualizada para un funcionamiento acorde a las necesidades de las personas que interactúan en el centro médico.

El Capítulo I, aborda toda la información relevante del Centro Médico de Orientación y Planificación Familiar CEMOPLAF Otavalo, Antecedentes, Situación actual, Definición del problema, Prospectiva del problema, Alcance y limitaciones del proyecto, y por último se presentan conclusiones y recomendaciones.

El Capítulo II, se describe todo lo referente al Proceso de Desarrollo Unificado Rational Rup, Definicion, Características , Aspectos a considerar, Ciclo de vida de un proyecto informatico, Estructuras del proceso(Dinámica y Estatica). Además se detalla las Herramientas de desarrollo, PHP, Frameworks y Patrones de Desarrollo Web, Gestores de Base de datos, y se termina el capítulo con conclusiones y recomendaciones.

El Capítulo III, se aborda el tema Desarrollo del sistema, iniciando por el Modelo de Negocio, Especificaciones de Requerimientos, Análisis del diseño, La implementación, Pruebas al sistema y por Ultimo el Despliegue en el que se realiza la Transferencia de Tecnología y Capacitación al personal que va a administrar el sistema.

Finalmente en el capítulo V, se termina con Conclusiones y Recomendaciones para el buen uso del sistema, ademas la bibliografia y anexos que apoyaron en la realizacion del presente trabajo de tesis.

## SUMMARY

This thesis entitled “Aplicación web para la Administración Online de Citas Médicas en el Centro Médico de Orientación y Planificación Familiar Cemoplaf-Otavalo; Utilizando el Patrón de Arquitectura MVC en php”, streamlines and standardizes the flow of information within each of the user management processes, booking appointments, medical management, secretaries, creating medical specialties. It was developed using free tools and high technology such as Codeigniter, MySQL, PHP, JavaScript and AJAX, while incorporating a comprehensive and sophisticated software development methodology called: RUP ((Rational Unified Process)).

The objective of the management system online dating is to have an efficient software tool that allows to obtain accurate and updated information for operation according to the needs of people interacting in the medical center.

Chapter I, addressing all relevant information Medical Center and Family Planning Guidance CEMOPLAF Otavalo, Background, Current Situation, Problem Definition, Forecasting the problem, scope and limitations of the project, and finally presents conclusions and recommendations.

Chapter II describes everything about the Rational Unified Process Development Rup, definition, characteristics, aspects to consider life cycle of a software project, process structures (dynamic and static), It also details the development tools, PHP, Frameworks and Patterns of Web Development, Database Managers, and the chapter ends with conclusions and recommendations.

Chapter IV, addresses the issue of system development, starting with the business model, requirements specification, design analysis, implementation, testing and finally system The deployment which performs the transfer of technology and training staff that is going to manage the system.

Finally in Chapter V, it ends with conclusions and recommendations for the proper use of the system, plus the bibliography and appendices that support the realization of this thesis.

# CAPÍTULO 1

## ANTECEDENTES, SITUACIÓN ACTUAL Y PROSPECTIVA



*Introducción*

*Antecedentes*

*Situación Actual*

*Definición del Problema*

*Prospectiva del Problema*

*Objetivos del Proyecto*

*Justificación del Proyecto*

*Alcance y Limitaciones del Proyecto*

## **1.1 INTRODUCCIÓN**

En la actualidad, el incremento de pacientes en las instituciones médicas ha requerido una mayor organización en la atención al cliente, por lo que se estableció como el primer proceso la reservación de citas médicas, optimizando el control del mismo. Sin embargo, existe la saturación y el colapso del proceso por lo que reduce la calidad del servicio, el paciente tiene que asistir al centro médico hacer largas filas para reservar su turno y luego asistir a su consulta, lo que se vuelve un trámite incómodo para el usuario ya que pierde tiempo y dinero en el proceso.

Esta problemática conlleva al uso de nuevas tecnologías que faciliten la reservación de una cita médica como son el uso del internet mediante un sitio online, el cual permitirá al paciente realizar reservaciones de citas médicas a cualquier hora y desde cualquier lugar, los 7 días de la semana, las 24 horas al día, entregando un mejor servicio y atención al cliente en cuanto a comodidad, eficiencia y gastos de recursos reducidos.

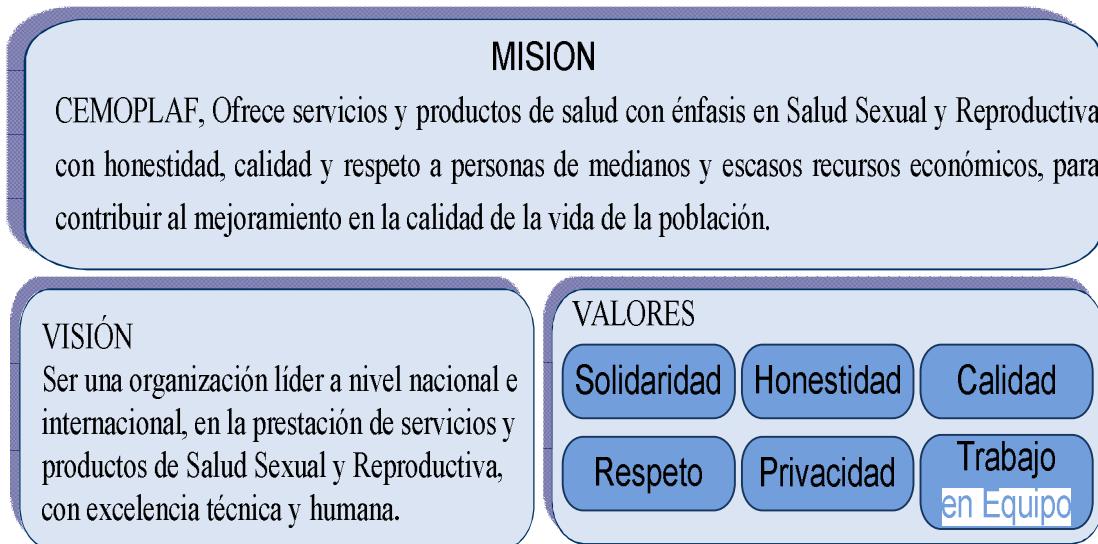
Es así que se plantea la necesidad de emprender un sistema de ADMINISTRACIÓN ONLINE DE CITAS MÉDICAS, como una alternativa eficiente, eficaz e innovadora de mejorar la prestación de servicios a los pacientes.

## **1.2 ANTECEDENTES**

CEMOPLAF, Centro Médico de Orientación y Planificación Familiar, es una organización no gubernamental, apolítica, y no lucrativa que presta servicios y productos de Salud con énfasis en Salud Sexual y Reproductiva a la población ecuatoriana.

Fue creada el 12 de Septiembre de 1974, con personería jurídica mediante Acuerdo Ministerial No. 3702, publicado en el Registro Oficial No. 637. *“Su experiencia a lo largo de estos años se fundamenta en el campo de la Salud Sexual, Reproductiva y Atención Materno Infantil, con una cobertura Importante a nivel País, Especialmente en áreas urbanas y rurales, con programas y proyectos integrales de salud, dirigidos especialmente a grupos prioritarios como madres, niños, adolescentes y familias de*

estatus<sup>1</sup> medios cuyas necesidades básicas son: salud sexual y reproductiva, seguridad alimentaria, seguridad anticonceptiva, educación en salud preventiva, protección del ambiente y desarrollo comunitario (REV01, Enero 2012)".



Fuente: Propia

**Figura 1.1.** Misión, Visión y Valores CEMOPLAF

### 1.3 SITUACIÓN ACTUAL

Actualmente CEMOPLAF Otavalo carece de un manejo eficiente de su información en épocas actuales en las que la publicación de contenidos y la prestación de servicios en línea en las dependencias públicas o privadas están en auge; el no contar con un sitio web que publique información importante y la necesidad de dar a conocer los servicios que actualmente posee, provoca el desconocimiento de la ciudadanía sobre los servicios médicos y de orientación que el dispensario ofrece, impidiendo la posibilidad de interactuar visitantes, usuarios y personal interno del centro.

Cemoplaf inició sus labores para promover la salud reproductiva en 1974 y contaba con once beneficiarias. En la actualidad, atiende a cerca de 500 mil pacientes al año. Después de tres décadas de trabajo en planificación familiar –junto con Aprofe, el Ministerio de Salud y otras organizaciones – los resultados son visibles. El

<sup>1</sup> **Estatus:** Posición social que un individuo ocupa dentro de una sociedad

porcentaje de mujeres en edad reproductiva que usan algún método anticonceptivo pasó de menos del 20% en la década del 70, al 73%, en el 2004.

Los estudios se han aplicado en campos como la divulgación de los métodos de planificación familiar entre la población indígena o la promoción de la salud reproductiva en los adolescentes. ([www01, 2005](#))

Una de las debilidades es la atención externa que CEMOPLAF Otavalo brinda a sus usuarios, en la que la solicitud de turno para ser atendido se lo debe realizar de manera personal en las instalaciones una hora antes de que médico llegue al consultorio, lo que implica que el usuario tiene que hacer largas filas para reservar su cita médica y luego regresar horas más tarde para poder acceder a la consulta de acuerdo al número de pacientes que el médico tenga asignado o esperar en las instalaciones a ser atendido, es por ello que las salas de espera siempre están llenas de usuarios; otra opción es solicitar su turno a través de una llamada telefónica que conlleva a mantener ocupada la línea del centro hasta concluir con la asignación de su pedido; siendo a veces imposible poder comunicarse.

CEMOPLAF, al ser una organización no lucrativa que presta servicios y productos de Salud con énfasis en Salud Sexual y Reproductiva a la población ecuatoriana de status medio, tiene una gran demanda en todos los servicios que presta, es por ello que cada vez hay personas que acuden a este lugar para buscar atención. CEMOPLAF tiene bien definidos sus objetivos y la atención presta a todos sus usuarios.

#### **1.4 DEFINICIÓN DEL PROBLEMA**

CEMOPLAF Otavalo posee una alta demanda de sus servicios profesionales en todas la área de atención, actualmente solo dispone de un sistema de citas por teléfono o personalmente, ocasionando demoras y retrasos en la asignación de citas médicas, teniendo como resultado la espera de los pacientes por más de una hora para ser atendidos; incluso en ocasiones la espera puede ser en vano debido a que el número de pacientes que un médico puede atender dentro de su horario laboral ya está saturado o el médico por algún contratiempo no pueda llegar a la consulta y retrase todas las atenciones posteriores, sin haber informado a los pacientes en

espera; por otra parte el *geleno*<sup>2</sup> del centro médico desconoce el número de pacientes y el motivo de la consulta de cada usuario a los que tiene que atender.

## **1.5 PROSPECTIVA DEL PROBLEMA**

Debido al desmedido incremento de usuarios en CEMOPLAF-Otavalo la falta de una aplicación Web dinámica en la cual pueda implementar sus servicios, difundir, controlar y administrar las citas e historial clínico de sus pacientes, repercutirá a que cada vez sea más difícil manejar el historial médico y consultas de los pacientes; y debido al alto número de pacientes que de alguna manera no fueron atendidos adecuadamente disminuya la calidad en sus servicios.

## **1.6 OBJETIVOS DEL PROYECTO**

### **Objetivo General**

- Desarrollar una Aplicación Web Para La Administración Online De Citas Médicas En El Centro Médico De Orientación Y Planificación Familiar CEMOPLAF-OTAVALO; Utilizando El Patrón De Arquitectura MVC en PHP.

### **Objetivos Específicos**

- Analizar y definir con claridad la información Fundamental de Centro Médico y establecer adecuadamente el proceso de solicitud de citas médicas a implementar.
- Aplicar como Metodología el Proceso Unificado de Rational, en la implementación de la Aplicación Web.
- Investigar los patrones de arquitectura y modelos en el desarrollo de Aplicaciones Web.
- Desarrollar una aplicación web que permita la administración de citas medicas según los perfiles de usuario establecidos
- Implementar y poner en correcto funcionamiento la aplicación web en el Internet.
- Transferir el conocimiento y capacitar al personal del Centro Médico, previo la

---

<sup>2</sup> **Geleno:** Hombre autorizado para ejercer la medicina.

## **1.7 JUSTIFICACIÓN DEL PROYECTO**

El Centro Médico de Orientación y Planificación CEMOPLAF posee una alta demanda de sus servicios profesionales, y actualmente no cuentan con un sistema informático online, que brinde de forma rápida, transparente y segura la administración de citas médicas. Es así que se plantea la necesidad de emprender un sistema de ADMINISTRACIÓN ONLINE DE CITAS MÉDICAS, como una alternativa eficiente, eficaz e innovadora de mejorar la prestación de sus servicios a los pacientes.

La facilidad y necesidad del uso del internet, hace que cada día más personas estén acostumbradas a realizar varias de sus tareas o actividades personales, tales como consultas y pagos de servicios básicos, transferencias bancarias, compras y ventas de productos, entre otros desde sus casas u oficinas con tan solo ingresar a la web.

Solicitar una Reserva o Cita Médica Online, ahorra tiempo y dinero, tanto para los profesionales como para los propios clientes o pacientes que pueden usar este servicio 24 horas al día, 7 días a la semana.

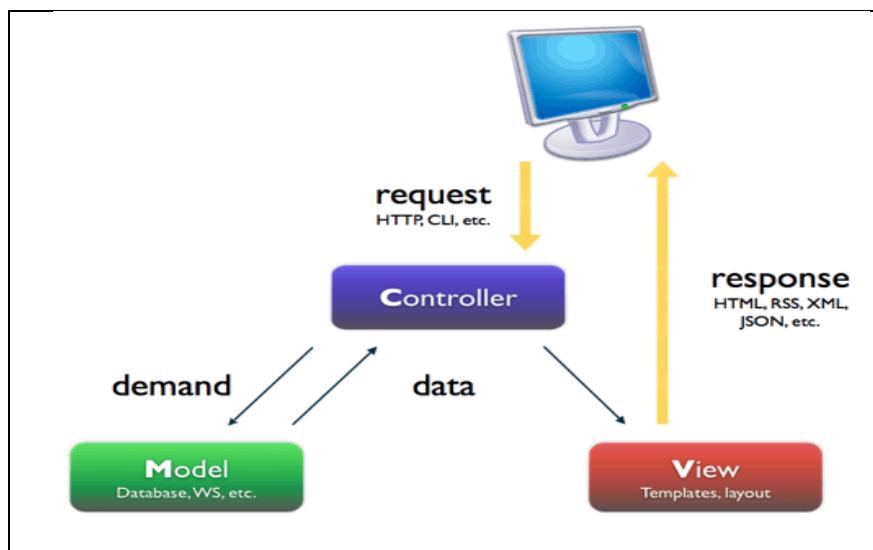
Los beneficios de acogerse a la Reserva de Citas Médicas Online, es su comodidad y rapidez. A través de Internet, en tiempo real, y eligiendo el día y la hora más conveniente con el médico idóneo, modificarla o incluso, llegado el caso, anularla. Y todo esto desde cualquier ordenador. Al mismo tiempo, nuestro sistema, proporciona un control exacto de las citas a través de una Agenda, y en el caso de Servicios de Salud, le proporciona al médico la posibilidad de mantenerse en contacto con su paciente a través e-mail u otras alternativas de comunicación en línea, estableciendo un vínculo entre pacientes y profesionales médicos.

### **1.7.1 Justificación Tecnológica**

El uso de tecnologías que permitan separar las diferentes capas presentes en una aplicación Web y la comprensión de su funcionamiento dentro de la aplicación a desarrollar, merece ser un tema de estudio; actualmente las aplicaciones se componen de 3 capas principales distribuidas de la siguiente manera:

- La capa del **modelo** que define la lógica de negocio (la base de datos pertenece a esta capa).

- La **vista** es lo que utilizan los usuarios para interactuar con la aplicación (los gestores de plantillas y páginas a las que el usuario accede pertenecen a esta capa).
- El **controlador** es un bloque de código que realiza llamadas al modelo para obtener los datos y se los pasa a la vista para que los muestre al usuario. Estos controladores frontales realmente delegan todo el trabajo en las acciones, y las agrupaciones lógicas de acciones se denominan módulos.

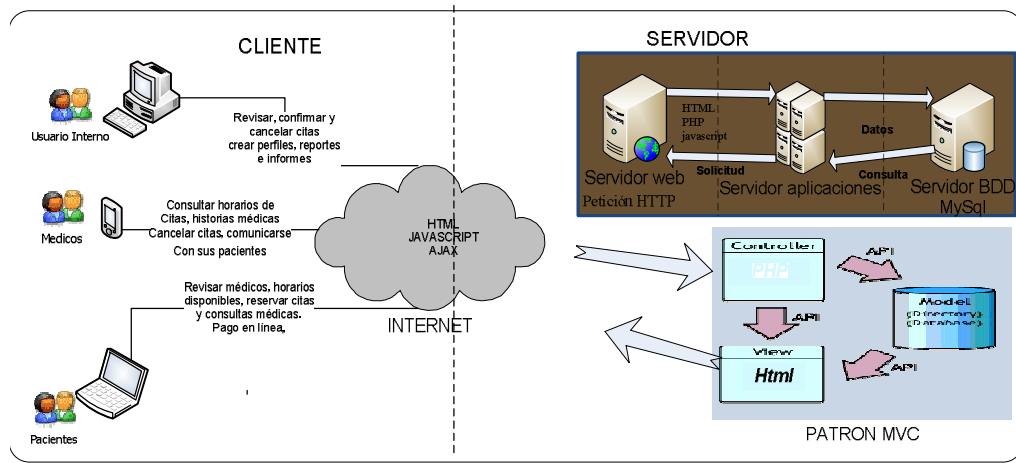


Fuente: (www07)

**Figura 1.2. Modelo Vista Controlador (MVC)**

La aplicación de las tendencias de software en el desarrollo de aplicaciones web en entornos *Open Source* nos brinda muchas ventajas, entre las que se destacan: independencia tecnológica, independencia entre los datos y la aplicación, libertad de uso y redistribución, ahorro económico, soporte y compatibilidad a largo plazo, formatos estándar, corrección más rápida y eficiente de fallos, métodos simples y unificados de gestión de software, y muchos colaboradores o comunidades dispuestas a ayudar.

Por tal motivo las tecnologías a utilizar en el desarrollo del sistema son PHP como lenguaje de servidor, bajo una arquitectura MVC, MySQL como servidor de base de datos, JavaScript como lenguaje de cliente y AJAX como intermediario entre el cliente y los datos del servidor, CodeIgniter como Framework de ayuda por ser increíblemente ligero que usa el diseño MVC y contiene una muy buena estructuración de sus librerías. Todo lo expuesto en este párrafo se representa con el siguiente diagrama.



Fuente: Propia

**Figura 1.3.** Arquitectura Planteada

### 1.7.2 Justificación Metodológica

El desarrollo de la presente aplicación web, se basará en la aplicación de la Metodología *RUP*, la misma que consta de las siguientes fases: Incepción, Elaboración, Construcción y Despliegue; y dentro del Desarrollo del Sistema se incluirá los principales artefactos necesarios para la correcta Documentación y Seguimiento del mismo.

Se utilizará esta metodología por su simplicidad, comunicación y la realimentación o reutilización del código desarrollado; dentro de todas las etapas del proyecto.

La documentación, a desarrollar en cada uno de los hitos del proyecto dentro de la Metodología se menciona a continuación:

- Visión
- Plan de Desarrollo
- Especificaciones y Requerimientos de Usuario
- Descripción de Casos de Uso
- Lista de Riesgos
- Modelo de Casos de Uso
- Modelo de Diseño
- Documentación de la Arquitectura
- Modelo de Implementación
- Plan de Pruebas

## 1.8 ALCANCE Y LIMITACIONES DEL PROYECTO

La Aplicación web deberá estar basada en el cumplimiento de los requisitos previamente establecidos por los Directivos del Centro Médico, la aplicación estará

enfocada principalmente en diseñar y desarrollar un sistema que posibilite al Centro Médico de Orientación y Planificación Familiar CEMOPLAF-Otavalo, administrar eficientemente las citas e identificación del historial clínico de sus pacientes.

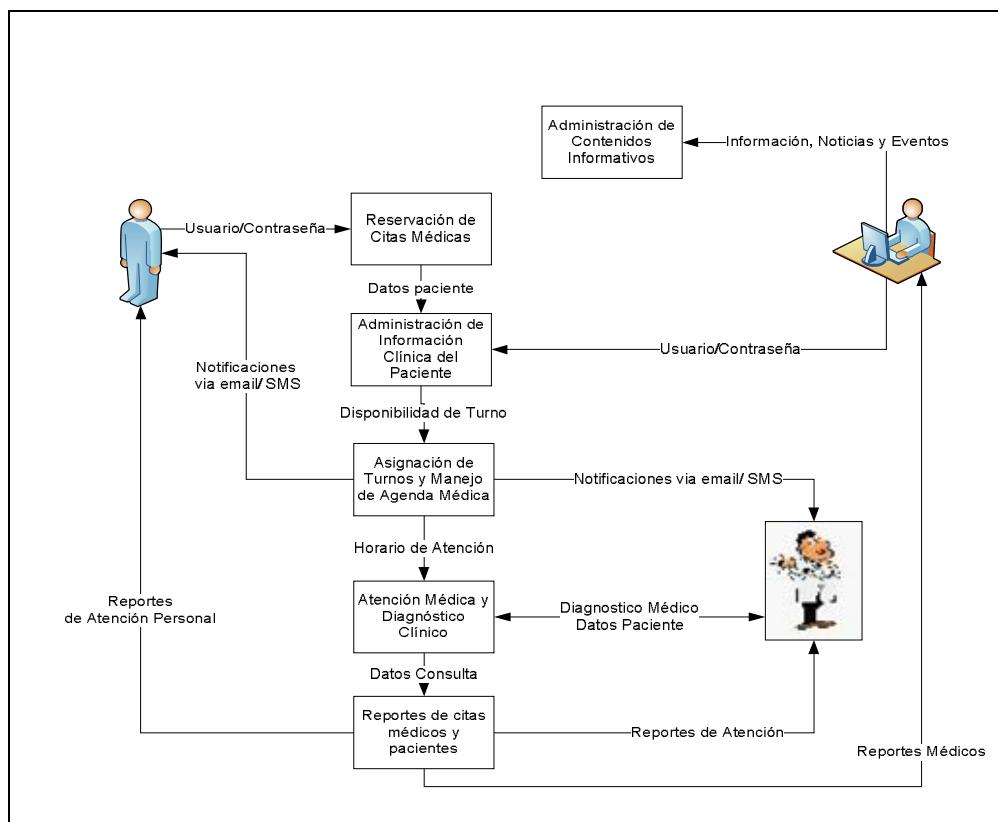
Algunas funciones principales que debe realizar este sistema son las siguientes:

- Gestionar correctamente las diferentes cuentas de usuario según los perfiles que se establezcan incrementando la seguridad e integridad de la información que maneja la aplicación.
- Permitir al Médico administrar la identificación de historiales clínicos de sus pacientes y obtener reportes instantáneos de sus citas realizadas, suspendidas, pendientes o eliminadas; independientemente del lugar donde se encuentre.
- Permitir al paciente hacer una cita online al centro médico así como su identificación de historial clínico de forma personalizada.

#### **1.8.1 Módulos Y Relaciones**

**Especificación de Funcionalidades:** En el Sistema Web con citas en línea de paciente y notificaciones a los médicos, se presentan las siguientes descripciones de las diferentes funcionalidades que tiene el sistema:

- *Página Web informativa:*
  - *Historia, Misión, Visión, Valores, Objetivos, Estructura, Autoridades*
  - *Eventos, Noticias y Artículos*
  - *Productos y Servicios*
  - *Pacientes:*
    - *Solicitud de citas médicas a pacientes frecuentes (que ya cuentan con una historia clínica).*
    - *Manejo de reservaciones de consulta*
    - *Revisión de datos médicos*
    - *Información de médicos disponibles*
  - *Médicos*
    - *Reporte de citas pendientes*
    - *Horarios de atención*
    - *Cancelación de citas*
    - *Notificaciones importantes (vía correo y SMS)*
  - *Personal Administrativo*
    - *Reporte de citas pendientes*
    - *Horarios de atención*
    - *Revisión de Solicitudes de citas*
    - *Cancelación de citas*
    - *Notificaciones importantes (vía correo y SMS)*



Fuente: Propia

**Figura 1.4.** Diagrama de Módulos y Relaciones

Todos los usuarios del sistema se deben autenticar con su respectiva clave para poder ingresar al sistema y así poder controlar y verificar su información personal; disponibilidad y horarios de atención.

En el Sistema Web, se debe tener en cuenta las seguridades es decir su clave o password con su código respectivo (número de historia médica o cedula de identidad); para no tener infiltrados no deseados que extraigan información confidencial. De esta manera se maneja la seguridad del sistema que es muy importante tener en cuenta para evitar daños a los usuarios.

# CAPÍTULO 2

## MARCO TEÓRICO



*Metodologías de Desarrollo*

*PHP (Hypertext Preprocessor)*

*Framework de Desarrollo WEB*

*Patrones de Diseño WEB*

*Gestores de Base de Datos*

*Herramientas de Desarrollo*

## **2.1 METODOLOGIA DE DESARROLLO**

### **2.1.1 INTRODUCCION**

Actualmente existe un gran apogeo del mercado de las Tecnologías de la Información, concretamente del mercado de productos de software, debido a que las empresas han notado la importancia de contar con un buen sistema que brinde soporte y automatice las actividades principales de la compañía.

RUP es una metodología en la que se describe quién, cómo, qué, en qué tiempo y qué actividades se van a desarrollar en el proyecto, el quién representa los distintos roles que puede desempeñar un individuo en la organización, el cómo se refiere a la unidad de trabajo que se asigna a un trabajador y el qué es la pieza de información utilizada por un proceso. Para el desarrollo satisfactorio de este proceso, es necesario cumplir con ciertas etapas, las mismas que en conjunto determinarán el ciclo de vida y el éxito de la aplicación. RUP hace uso de cuatro etapas en su metodología descritas de la siguiente manera: una fase de concepción, una fase de elaboración, una fase de construcción y una última de transición. Dentro de cada una de ellas se realizan varias iteraciones dependiendo de la dimensión del proyecto.

La metodología RUP es más apropiada para proyectos grandes (Aunque también pequeños), dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas. En proyectos pequeños, es posible que no se puedan cubrir los costos de dedicación del equipo de profesionales necesarios.

Los orígenes de RUP se remontan al modelo espiral original de *Barry Boehm*. Ken Hartman, uno de los contribuidores claves de RUP colaboró con Boehm en la investigación. En 1995 Rational Software compró una compañía sueca llamada Objectory AB, fundada por Ivar Jacobson, famoso por haber incorporado los casos de uso a los métodos de desarrollo orientados a objetos. El Rational Unified Process fue el resultado de una convergencia de Rational Approach y Objectory (el proceso de la empresa Objectory AB). El primer resultado de esta fusión fue el Rational Objectory Process, la primera versión de RUP, fue puesta en el mercado en 1998, siendo el arquitecto en jefe Philippe Kruchten. ([www03](#)).

### **2.1.2 RUP (*Proceso Unificado Rational*)**

El Proceso Unificado de Rational (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado *UML*, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos (www03).

### **2.1.3 Características RUP**

Los autores de RUP destacan que el proceso de software propuesto tiene tres características esenciales:

#### **a) Dirigido/Manejado por Casos de Uso**

Los casos de uso reemplazan la antigua especificación funcional tradicional y constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo, incluyendo el diseño, la implementación y las pruebas del sistema (www02).

En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba.

#### **b) Centrado en la Arquitectura**

La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo. (LIB01, 2000)

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, *protocolos*, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema.

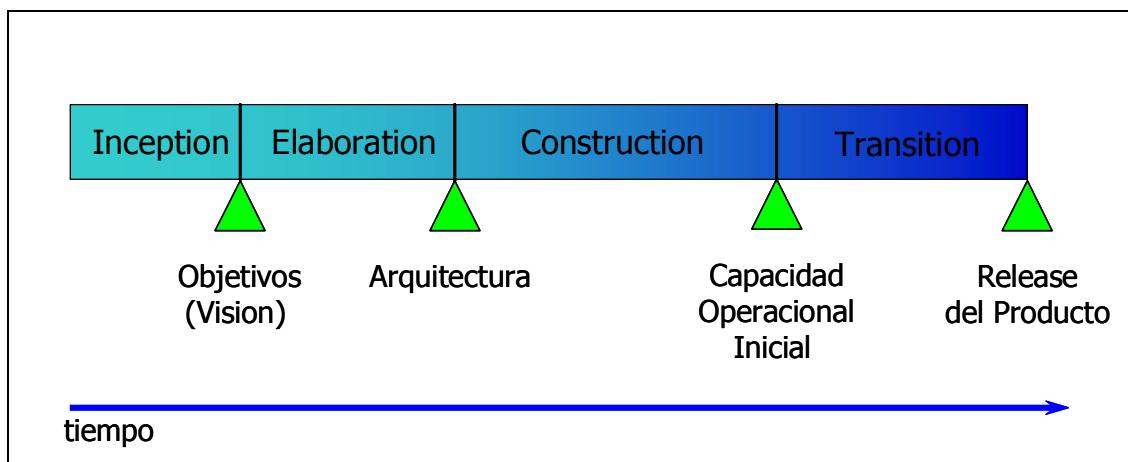
### c) Interactivo e Incremental

Para hacer más manejable un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un mini-proyecto, cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo.

#### 2.1.4 ESTRUCTURA DINÁMICA DEL PROCESO FASES E INTERACCIONES

RUP se repite a lo largo de una serie de ciclos que constituyen la vida de un producto. Cada ciclo concluye con una generación del producto para los clientes. Cada ciclo consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada fase se subdivide a la vez en iteraciones, el número de iteraciones en cada fase es variable.

Cada fase se concluye con un hito bien definido, un punto en el tiempo en el cual se deben tomar ciertas decisiones críticas y alcanzar las metas clave antes de pasar a la siguiente fase, ese hito principal de cada fase se compone de hitos menores que podrían ser los criterios aplicables a cada iteración. Los hitos para cada una de las fases son: Inicio - *Lifecycle Objectives*, Elaboración - *Lifecycle Architecture*, Construcción - *Initial Operational Capability*, Transición - *Product Release*. Las fases y sus respectivos hitos se ilustran en la Figura 2.1.



Fuente: ([www05](http://www05))

**Figura 2.1.** Fases e hitos en RUP

### **a) Incepción**

Se obtiene una visión inicial del producto así como su alcance, además se identifican los principales casos de uso.

Antes de iniciar el proyecto es conveniente plantearse algunas cuestiones: el objetivo, la factibilidad, construir o adaptar y el costo que involucra. La fase de inicio trata de responder a estas preguntas y a otras más. Sin embargo una estimación precisa debe ser la captura de todos los requisitos. Explorar el problema para decidir si continuar o concluir. Generalmente no debe tardar más de una semana.

Los productos de la fase de inicio pueden ser: (LIB04, 1998).

- Un documento de visión: Una visión general de los requerimientos del proyecto, características clave y restricciones principales.
- Modelo inicial de Casos de Uso (10-20% completado).
- Un glosario inicial: Terminología clave del dominio.
- El caso de negocio.
- Lista de riesgos y plan de contingencia.
- Modelo de negocio, si es necesario
- Prototipos exploratorios para probar conceptos o la arquitectura candidata.

Bajo estos conceptos en el proyecto se establecen los siguientes aspectos a considerar que determinaran la factibilidad del mismo los cuales se encuentran detallados en el documento de Visión (**Anexo 2**).

### **b) Elaboración**

En esta fase se planifica las actividades y el equipo de trabajo del proyecto, se identifican las necesidades y el diseño de la arquitectura. El propósito de la fase de elaboración es analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del proyecto y eliminar los mayores riesgos.

En esta fase se construye un prototipo de la arquitectura, que debe evolucionar en iteraciones sucesivas hasta convertirse en el sistema final. Este prototipo debe contener los casos de uso críticos identificados en la fase de inicio. También debe demostrarse que se han evitado los riesgos más graves, bien con este prototipo, bien con otros de usar y tirar.

Al terminar deben obtenerse los siguientes productos (LIB05, 2002).

- Modelo de Casos de Uso completo
- Requisitos adicionales.

- Descripción de la arquitectura software.
- Lista de riesgos y caso de negocio revisados.
- Plan de desarrollo para el proyecto.
- Un plan de desarrollo actualizado que especifique el proceso a seguir.

Todo esto se resume en el modelo de Negocio (**Figura 3.1**) lista de actores del sistema (**Figura 3.3**) y los Casos de Uso (**Figuras 3.2 – Figuras 3.13, Tablas 3.4 – 3.26**) detallados en el siguiente capítulo.

### **c) Construcción**

Comprende el desarrollo mismo del producto hasta la entrega al usuario final. La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones. Durante esta fase todas los componentes, características y requisitos, que no lo hayan sido hecho hasta ahora, han de ser implementados, integrados y probados, obteniéndose una versión del producto que se pueda poner en manos de los usuarios (una *versión beta*).

El énfasis en esta fase se pone controlar las operaciones realizadas, administrando los recursos eficientemente, de tal forma que se optimicen los costes, los calendarios y la calidad.

Los productos de la fase de construcción (LIB04, 1998) deben ser:

- Modelos completos (Casos de Uso, Análisis, Diseño, Despliegue e Implementación)
- Arquitectura íntegra (mantenida y mínimamente actualizada)
- Plan del proyecto para la fase de transición
- Manual inicial de usuario (con suficiente detalle)
- Prototipo operacional – beta
- Caso de negocio actualizado

Todo esto se resume en los manuales Técnico Funcional (**Anexo 4**), Manual de Usuario, (**Anexo 3**).

### **d) Transición y Despliegue**

Esta fase comprende la instalación del producto a los usuarios y la formación de los mismos, en ocasiones suelen surgir nuevos requisitos para el desarrollo.

La finalidad de la fase de transición es poner el producto en manos de los usuarios finales, para lo que típicamente se requerirá desarrollar nuevas versiones actualizadas del producto, completar la documentación, entrenar al usuario en el

manejo del producto y, en general, tareas relacionadas con el ajuste, configuración, instalación y usabilidad del producto.

En <sup>(LIB01, 2000)</sup> se citan algunas de las cosas que puede incluir esta fase:

- Prueba de la versión para validar el nuevo sistema frente a las expectativas de los usuarios.
- Funcionamiento paralelo con los sistemas legados que están siendo sustituidos por nuestro proyecto, si los hay.
- Entrenamiento de los usuarios y técnicos de mantenimiento.

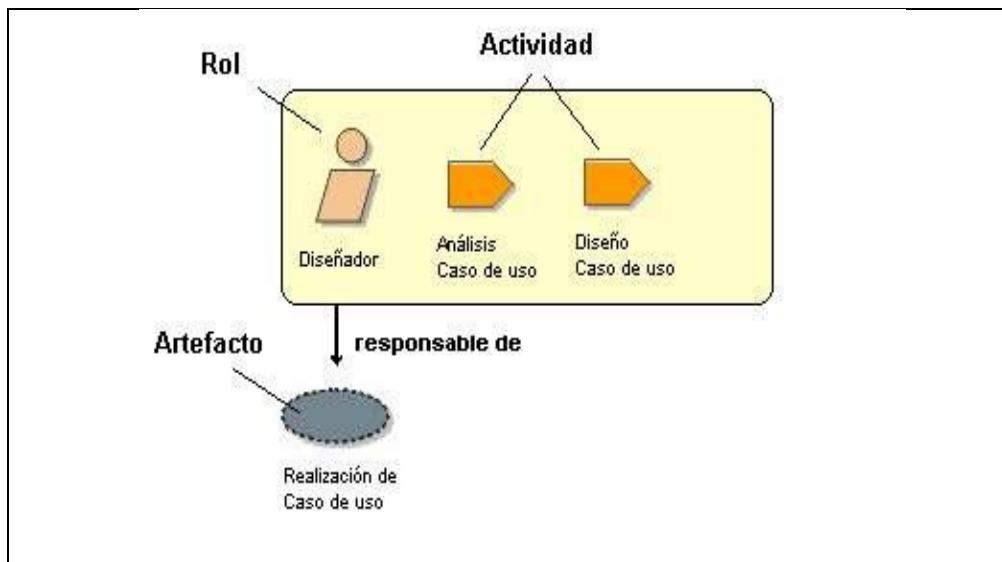
Los productos de la fase de transición según <sup>(LIB04, 1998)</sup> son:

- Prototipo operacional
- Documentos legales
- Caso de negocio completo
- Descripción de la arquitectura completa y corregida
- Las iteraciones de esta fase irán dirigidas normalmente a conseguir una nueva versión.

Todo esto se resume en el plan de pruebas y Evaluación de resultados (**Tabla 3.23 – 3.43**), dentro del capítulo de Desarrollo del sistema; la transición y transferencia de conocimientos se detalla en el manual Técnico Funcional.

#### **2.1.5 ESTRUCTURA ESTÁTICA DEL PROCESO, ROLES, ACTIVIDADES, ARTEFACTOS Y FLUJOS DE TRABAJO**

Un proceso de desarrollo de software define quién hace qué, cómo y cuándo. RUP define cuatro elementos: los roles, que responden a la pregunta ¿Quién?, las actividades que responden a la pregunta ¿Cómo?, los productos, que responden a la pregunta ¿Qué? y los flujos de trabajo de las disciplinas que responde a la pregunta ¿Cuándo?.



Fuente: (www05)

**Figura 2.2.** Relación entre roles, actividades, artefactos

#### • Roles

Un rol define el comportamiento y responsabilidades de un individuo, o de un grupo de individuos trabajando juntos como un equipo. Una persona puede desempeñar diversos roles, así como un mismo rol puede ser representado por varias personas.

#### • Actividades

Una actividad en concreto es una unidad de trabajo que una persona que desempeñe un rol puede ser solicitado a que realice. Las actividades tienen un objetivo concreto, normalmente expresado en términos de crear o actualizar algún producto.

#### • Artefactos

Un producto o artefacto es un trozo de información que es producido, modificado o usado durante el proceso de desarrollo de software. Los productos son los resultados tangibles del proyecto, las cosas que va creando y usando hasta obtener el producto final.

#### • Flujos de Trabajo

Un flujo de trabajo es una relación de actividades que nos producen unos resultados observables. A continuación se dará una explicación de cada flujo de trabajo. En RUP se definen nueve flujos de trabajo distintos, separados en dos grupos como se muestra en la siguiente tabla.

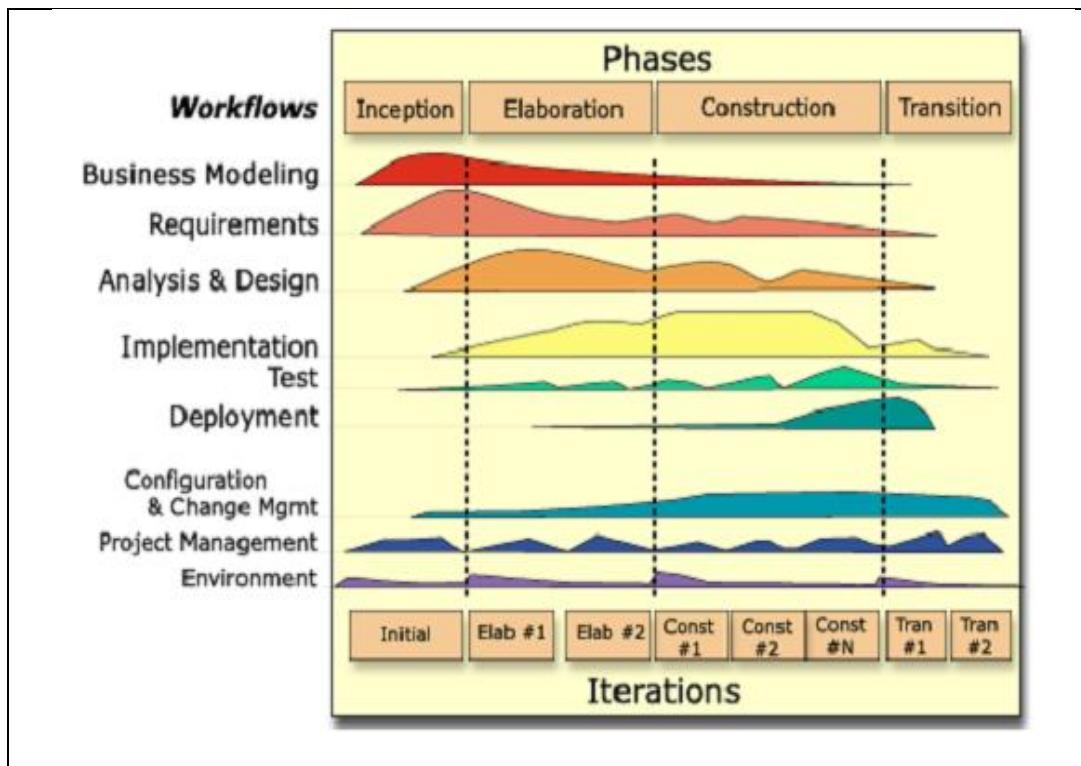
FLUJOS DE TRABAJO DE: INGENIERIA	FLUJOS DE TRABAJO DE: SOPORTE
Modelado del Negocio Requerimientos Análisis y Diseño Implementación Pruebas Despliegue	Gestión del Proyecto Gestión del cambio y configuraciones Ambiente o Entorno

Fuente: Propia

**Tabla 2.1.** Flujos de Trabajo

### 2.1.6 INTEGRACION DE FASES Y ACTIVIDADEDES EN RUP

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en los distintas actividades. En la Figura 2.3 se muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.



Fuente: ([www05](http://www05))

**Figura 2.3.** Esfuerzo en actividades según fase del proyecto

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la

eliminación de los riesgos críticos, y al establecimiento de una baseline de la arquitectura.

Durante la fase de inicio las iteraciones ponen mayor énfasis en actividades modelado del negocio y de requisitos.

En la fase de elaboración, las iteraciones se orientan al desarrollo de la baseline de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la baseline de la arquitectura.

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Para cada iteración se selecciona algunos Casos de Uso, se refina su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan tantas iteraciones hasta que se termine la implementación de la nueva versión del producto.

En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

Como se puede observar en cada fase participan todas las disciplinas, pero que dependiendo de la fase el esfuerzo dedicado a una disciplina varía.

## **2.2 PHP (Hypertext Preprocessor)**

Es un lenguaje de programación interpretado en el servidor, con una sintaxis similar a la de C++ o JAVA; Creado originalmente en el año de 1994 pero que actualmente es ampliamente usado en entornos de desarrollo web por su facilidad de uso, su integración perfecta con *ficheros HTML* y su versatilidad de uso en diferentes Sistemas Operativos.

Tanto es su expansión, que se calcula su uso en torno a más de 20 millones de sitios web y un millón de servidores en todo el mundo . Aunque el lenguaje se puede usar para realizar cualquier tipo de programa, es en la generación dinámica de páginas web donde ha alcanzado su máxima popularidad; suele incluirse incrustado en páginas con otras tecnologías.

## 2.2.1 Aspectos a considerar

Algunas de sus características son:

- Es un lenguaje libre. Puede descargarse de <http://www.php.net>.
- Está disponible para muchos sistemas (GNU/Linux, Windows, UNIX).
- Tiene una extensa documentación oficial en varios idiomas
- Existen multitud de extensiones: para conectar con bases de datos, generar documentos PDF y generar dinámicamente páginas en Flash.
- Al ejecutarse en el servidor, se puede usar todo tipo de máquinas con todo tipo de sistemas operativos.

La característica más potente y destacable es su soporte para una gran cantidad de bases de datos, las siguientes bases de datos están soportadas actualmente:

- dBase InterBase PostgreSQL
- Empress FrontBase Solid
- FilePro MySQL Sybase
- IBM DB2 MySQL Velocis
- Informix ODBC Unix dbm

Otro aspecto, es que PHP debe trabajar conjuntamente con HTML y otros diferentes tipos de recursos para la implementación de un sitio dinámico. Por sí solo, PHP no aporta todos los elementos con que un sitio debe contar. ([www08, 2005](#)) ; soporta también el uso de otros servicios de mensajería, correo, encriptación de la información y seguridad con el uso de varios protocolos como *IMAP*, *SNMP*, *NNTP*, *POP3*, *HTTP* y derivados. ([www13, 2000](#)); que son utilizados para el envío y notificación de información, ya sea mediante correo electrónico, mensajería instantánea, generación de archivos de texto, entre otros.

Un servidor remoto brinda alojamiento a un sitio desarrollado en PHP el cual debe contar, con una configuración que le permita interpretar y ejecutar las páginas dinámicas generando los resultados en páginas *HTML* estándar. Estas páginas *HTML* serán enviadas al cliente para que su navegador las ejecute sin necesidad de algún software especial; tal como lo muestra la siguiente que de forma muy sencilla establece las peticiones del cliente de una página dinámica a través de su *URI* a un servidor de Internet. Este localiza la página y con su configuración interpreta y ejecuta la página dinámica; generando páginas *HTML* que son enviadas como respuesta.



Fuente: (www08, 2005)

**Figura 2.4.** Solicitud de una página dinámica a través de su URI

### 2.2.2 Arquitectura y Servicios

PHP permite una arquitectura Cliente – Servidor de 3 y más capas; para el manejo y el intercambio de información; la ejecución de PHP, inspecciona las consultas enviadas al servidor de Base de Datos para administrar correctamente sus recursos, y transaccionar entre el Servidor Web que es el encargado de toda la interpretación de los archivos PHP y su generación a Ficheros HTML; con los datos obtenidos en las consultas.

Los diferentes servicios ofrecidos por PHP, están relacionados con el objetivo del proyecto o sistema a desarrollar, su gran cantidad de bibliotecas (librerías de mail, conexión a Bases de Datos, manejo de imágenes) y funciones propias desarrolladas o disponibles en la web de otros autores por su tendencia Open Source ([sourceforge.net](http://sourceforge.net)); permiten desarrollar aplicaciones lo suficientemente robustas y de gran escala que están disponibles a través de la Web, si a todo esto se suma el gran auge de sistemas maestrales de contenidos CMS y frameworks de trabajo basados en PHP, lo convierten en una gran alternativa en el desarrollo de software orientado a la Web.

## 2.3 FRAMEWORKS DE DESARROLLO WEB

### 2.3.1 INTRODUCCION

Los frameworks para PHP, son un conjunto de archivos PHP que vienen preparados con toda la estructura necesaria para desarrollar varios tipos de proyectos .

Hay muchos Frameworks de PHP; cada uno orientado a un tipo de proyecto, por lo que elegir el adecuado puede ahorrar una gran cantidad de trabajo, ademas se debe tomar en cuenta que framework se ajusta mas a las necesidades de un proyecto de Desarrollo.

Desarrollar un proyecto en PHP, muchas veces requiere escribir todo el código desde cero, lo que no es muy práctico. Para ello se puede de aplicar e implementar códigos ya desarrollados para funciones frecuentes (Insertar, Recupera Datos, Autenticarse en el sistema, etc) que se encuentran disponibles en los frameworks, que hará más óptimo el trabajo y centralizar el esfuerzo específicamente en la lógica de la aplicación.

### **2.3.2 Framework PHP CodeIgniter**

Es un framework ligero que usa el Patrón de Diseño MVC y contiene una muy buena estructuración de sus componentes. Permite desarrollar proyectos mucho más rápido ya que cuenta con un rico conjunto de bibliotecas para tareas comunes, así como una interfaz sencilla y una estructura lógica para acceder a las mismas, permite enfocarse creativamente en el proyecto al minimizar la cantidad de código necesario para una tarea dada; otras aspectos importantes son:

- Está liberado bajo licencias open source del estilo *Apache/BSD*.
- Liviano; el núcleo del sistema sólo requiere algunas bibliotecas muy pequeñas; Las bibliotecas adicionales se cargan dinámicamente.
- Las *URLs* generadas son claras y amigables ya que usa un enfoque basado en segmentos: *ejemplo.com/noticias/articulo/345*
- Amplia compatibilidad con cuentas estándar de alojamiento que corren una variedad de versiones de PHP y configuraciones.
- Casi no necesite configuración.
- Compatibilidad con una amplia variedad de servidores y configuraciones
- Flexibilidad, no obliga a tener una determinada estructura de tablas, nombres de campos, ni adherirse a una forma de programar concreta.
- Evita la complejidad, favoreciendo las soluciones simples.
- Tiene una documentación clara y completa.
- Sus requerimientos de ejecución son mínimos: PHP versión 5.1.6 o más reciente; Bases de datos soportadas actualmente como MySQL (4.1+), MySQLi, MS SQL, Postgres, Oracle, SQLite, y ODBC.

Dentro de los inconvenientes que son mínimos podemos destacar:

- Curva de aprendizaje: necesidad de aprender nuevas funciones, estructuras y métodos de programación.
- Dificultad para adaptar el código escrito en PHP tradicional.

### **2.3.3 Historia y Evolución CodeIgniter**

CodeIgniter fue desarrollado originalmente por Rick Ellis (*CEO* de EllisLab). El framework se escribió para obtener buen rendimiento en el mundo real, donde muchas de las bibliotecas de clases, *helpers*, y subsistemas se tomaron prestados del código base de *ExpressionEngine*. Actualmente, el Equipo de Desarrollo de ExpressionEngine lo desarrolla y mantiene.

Actualmente la última versión estable a la fecha de elaboración del presente proyecto es la versión 2.2; que sirve de base para la elaboración, diseño e implementación de la parte aplicativa de este trabajo.

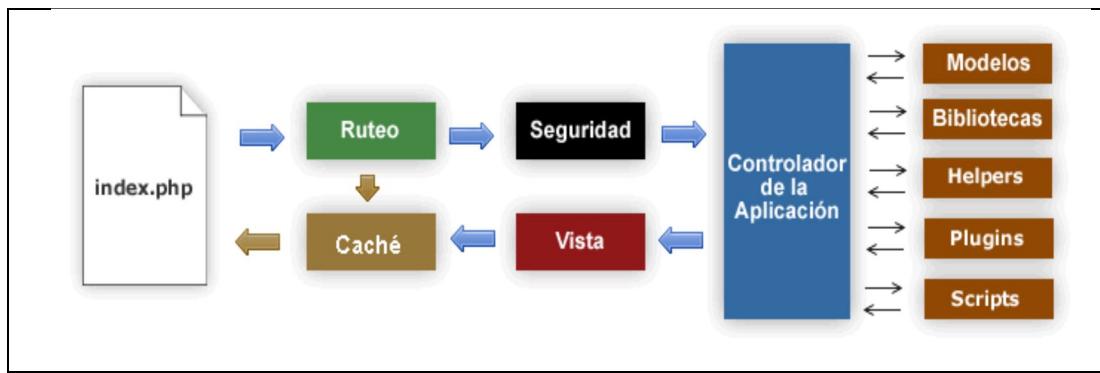
### **2.3.4 Funcionalidades de CodeIgniter**

Dentro de los componentes disponibles del framework se encuentran varias funcionalidades para los diferentes escenarios posibles en el desarrollo de una aplicación Web a mediana o gran escala; por el momento, la lista de funcionalidades principales de CodeIgniter son:

- Sistema basado en Modelo-Vista-Controlador
- Amplia biblioteca de funciones "helper"
- Clases de base de datos con soporte para varias plataformas
- Soporte para base de datos con Active Record
- Validación de datos y formularios
- Seguridad y filtrado
- Administración de sesiones
- Clase para subir archivos
- Encriptación de datos
- Clase para Calendarios
- Clase para Agente del Usuario
- Clase para codificación Zip
- Biblioteca XML-RPC
- URLs amigables para los motores de búsqueda
- Ruteo URI flexible

### **2.3.5 Flujo de la Aplicación de CodeIgniter**

El siguiente gráfico ilustra como los datos fluyen a través del sistema:



Fuente: (www12, 2012)

**Figura 2.5.** Flujo de datos a través del sistema

- El index.php sirve como el controlador frontal, inicializando los recursos básicos que necesita CodeIgniter para ejecutar.
- El Ruteador examina la solicitud HTTP para determinar que debería hacer con ella.
- Si existe el archivo de caché, se lo envía directamente al navegador, sin pasar por la ejecución normal del sistema.
- Seguridad. Antes que se cargue el controlador de la aplicación, por razones de seguridad se filtran la solicitud HTTP y cualquier otro dato enviado por el usuarios.
- El controlador carga el modelo, las bibliotecas del núcleo, helpers, y cualquier otro recurso requerido para procesar una solicitud específica.
- La Vista terminada se procesa y se envía al navegador para que se pueda ver. Si el caché está habilitado, la vista se cachea primero para que las siguientes solicitudes que la necesiten puedan ser servidas.

## 2.4 PATRONES DE DISEÑO WEB

### 2.4.1 INTRODUCCIÓN

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Un patrón de diseño resulta ser una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

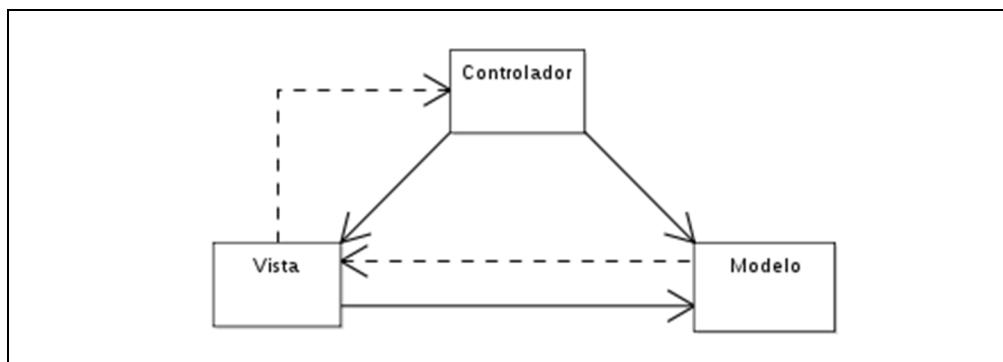
El creciente auge en el desarrollo de aplicaciones Web que se vive actualmente nos orienta a buscar un Patrón de Desarrollo altamente extensible, que permita futuras

modificaciones posteriores, cambios o adaptaciones con el menor esfuerzo posible; esto nos lleva a plantear dentro del presente trabajo uno de los más utilizados como los es MVC, que sirve de base para muchas aplicaciones disponibles hoy en día; y que se complementa y adapta fácilmente con Codeigniter.

#### **2.4.2 Patrón de Arquitectura MVC**

El patrón de Arquitectura Modelo-Vista-Controlador. MVC es un enfoque de software que separa la lógica de la aplicación de la presentación. En la práctica, permite a las páginas web contener el mínimo código ya que la presentación está separada del código PHP y de la recuperación e Inserción de Información en la Base de Datos.

Se originó en la comunidad Smalltalk para implementar interfaces de usuario en los que las responsabilidades están bien distribuidas entre distintas partes del diseño (componentes); dividiendo la aplicación de acuerdo a las responsabilidades dentro de la misma; entre la lógica de negocio, la gestión de eventos y la presentación.



Fuente: Propia

**Figura 2.6.** Arquitectura MVC

#### **2.4.3 Diseño y Arquitectura**

- **Modelo**

El Modelo representa las estructuras de datos; es decir toda la información con la que opera la aplicación; Gestionar el comportamiento y los datos del dominio; típicamente las clases del modelo contendrán funciones que ayudan a devolver, insertar y actualizar información de su base de datos, independientemente de la Base de Datos a utilizar o posibles cambios.

Encapsular el modelo de una aplicación en componentes facilita la depuración, mejora la calidad y favorece la reutilización de código, y los cambios que el sistema pudiera sufrir a futuro.

- **Vista**

La Vista es la información que se presenta al usuario. La vista gestiona la presentación de la información de la aplicación; todo lo relativo a la interfaz de usuario, los datos necesarios para que el usuario pueda seguir interactuando con la aplicación.

Una vista será normalmente una página web, La vista representa la lógica de presentación de la aplicación. Los componentes de la vista extraen el estado actual del sistema del modelo y proporcionan la interfaz de usuario para el protocolo que se está usando.

Como parte de la generación la vista debe presentar al usuario el conjunto de eventos que puede generar en ese momento en CodeIgniter, una vista también puede ser un fragmento de página como el encabezado o pie de página.

También puede ser una página RSS, o cualquier otro tipo de "página".

- **Controlador**

Es el indicado a responder a los eventos invocados desde la vista; y a su vez llamar a los métodos correspondientes el modelo para procesar y producir las respuestas necesarias con los datos adecuados para la interacción de la aplicación.

El Controlador sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la solicitud HTTP y generar una página web.

El controlador es responsable de recibir los eventos, determinar el procesador del evento, invocar al procesador y finalmente provocar la generación de la vista apropiada.; deben realizar las siguientes tareas:

- Control de la seguridad.
- Identificación de eventos.
- Preparar el modelo.
- Procesar el evento.
- Manejar los errores.
- Provocar la generación de la respuesta.

CodeIgniter tiene un enfoque bastante poco estricto de MVC, ya que los Modelos no son obligatorios. Si no necesita la separación añadida o encuentra que mantener modelos requiere más complejidad de la que quiere, puede ignorarlos y construir su aplicación mínimamente usando Controladores y Vistas.

CodeIgniter también le permite incorporar sus propios scripts, o inclusive desarrollar bibliotecas del núcleo para el sistema, permitiéndole trabajar en una forma que tenga más sentido para Ud. ([www12.2012](http://www12.2012))

## **2.5 GESTORES DE BASES DE DATOS**

Los sistemas de gestión de bases de datos *SGBD* son un tipo de software específico dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

El objetivo principal de las bases de datos es el de unificar los datos que se manejan y los programas o aplicaciones que los manejan. Anteriormente los programas se codificaban junto con los datos, es decir, se diseñaban para la aplicación concreta que los iba a manejar, lo que desembocaba en una dependencia de los programas respecto a los datos, ya que la estructura de los ficheros va incluida dentro del programa, y cualquier cambio en la estructura del fichero provocaba modificar y recompilar programas. Además, cada aplicación utiliza ficheros que pueden ser comunes a otras de la misma organización, por lo que se produce una redundancia de la información, que provoca mayor ocupación de memoria, laboriosos programas de actualización (unificar datos recogidos por las aplicaciones de los diferentes departamentos), e inconsistencia de datos (no son correctos) si los datos no fueron bien actualizados en todos los programas.

### **2.5.1 Administración de Recursos**

Con las bases de datos, se busca independizar los datos y las aplicaciones, es decir, mantenerlos en espacios diferentes. Los datos residen en memoria y los programas mediante un sistema gestor de bases de datos, manipulan la información. El sistema gestor de bases de datos recibe la petición por parte del programa para manipular los datos y es el encargado de recuperar la información de la base de datos y devolvérsela al programa que la solicitó

Un Motor de Base de Datos Libre, que se adapte al framework y facilmente soportado en el alojamiento Web de este proyecto es MySQL; una base de datos muy rápida en la lectura en aplicaciones web debido a la baja concurrencia en la modificación de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Es importante el desempeño para detectar y corregir errores SQL y de programación, que facilmente se lo puede determinar con las herramientas adecuadas.

### **2.5.2 Aspectos a considerar**

Aunque hay multitud de aplicaciones para la Gestión de Bases de Datos diferentes en características (libres, de pago, código abierto, cerrado, etc.) , existen aspectos comunes en todos ellos:

- Definiciones de esquemas y vistas.
- Manipulan los datos siguiendo las órdenes de los usuarios.
- Cuidan que se respete la seguridad e integridad de los datos.
- Permiten definir usuarios y las restricciones de acceso para cada uno de ellos.
- Controlan la concurrencia y las operaciones asociadas a la recuperación de los fallos.

Existen distintos objetivos que deben cumplir los SGBD:

**Abstracción de la información.** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario.

**Independencia.** La independencia de los datos capaz de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.

**Consistencia.** En aquellos casos en los que no se ha logra eliminar la redundancia, es necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea. Por otra parte, la base de datos representa una realidad que tiene determinadas condiciones.

**Seguridad.** La información almacenada en una base de datos puede llegar a tener un gran valor por lo que se debe garantizar que esta información se encuentra segura con varios niveles de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías.

**Tiempo de respuesta.** Minimizar el tiempo que el Gestor de Base de Datos demora en proporcionar la información solicitada y en almacenar los cambios realizados..

### **2.5.3 Arquitectura y Servicios**

En 1975, el comité **ANSI-SPARC** propuso una arquitectura de tres niveles para los sistemas de bases de datos, que resulta muy útil a la hora de conseguir estas tres características.

El objetivo de la arquitectura de tres niveles es separar los programas de aplicación de la base de datos física. En esta arquitectura, el esquema de una base de datos se define en tres niveles de abstracción distintos:

- En el nivel interno la estructura física de la base de datos mediante un esquema interno. Este esquema se especifica mediante un modelo físico y describe todos los detalles para el almacenamiento de la base de datos, así como los métodos de acceso.
- La mayoría de los SGBD no distinguen del todo los tres niveles. Algunos incluyen detalles del nivel físico en el esquema conceptual. En casi todos los SGBD que se manejan vistas de usuario, los esquemas externos se especifican con el mismo modelo de datos que describe la información a nivel conceptual, aunque en algunos se pueden utilizar diferentes modelos de datos en los niveles conceptuales y externos.
- En el nivel conceptual se describe la estructura de toda la base de datos para una comunidad de usuarios (todos los de una empresa u organización), mediante un esquema conceptual. Este esquema oculta los detalles de las estructuras de almacenamiento y se concentra en describir entidades, atributos, relaciones, operaciones de los usuarios y restricciones. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar el esquema.

En el nivel externo se describen varios esquemas externos o vistas de usuario. Cada esquema externo describe la parte de la base de datos que interesa a un grupo de usuarios determinado y oculta a ese grupo el resto de la base de datos. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar los esquemas.

### **2.5.4 Mysql**

MySQL es un sistema gestor de bases de datos muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros Gestores de Bases de Datos del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento

precisamente por su facilidadde uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo.

Las características de Mysql son:

- Está desarrollado en C/C++.
- Se distribuyen ejecutables para cerca de diecinueve plataformas diferentes.
- La API se encuentra disponible en C, C++, Java, Perl, PHP, Python, Ruby.
- Está optimizado para equipos de múltiples procesadores.
- Es muy destacable su velocidad de respuesta.
- Se puede utilizar como cliente-servidor o incrustado en aplicaciones.
- Cuenta con un rico conjunto de tipos de datos.
- Soporta múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento.
- Su administración se basa en usuarios y privilegios.
- Se tiene constancia de casos en los que maneja cincuenta millones de registros, sesenta mil tablas y cinco millones de columnas.
- Sus opciones de conectividad abarcan TCP/IP, sockets UNIX y sockets NT, además de soportar completamente ODBC.
- Los mensajes de error pueden estar en español y hacer ordenaciones correctas con palabras acentuadas o con la letra 'ñ'.
- Es altamente confiable en cuanto a estabilidad se refiere.
- Para todos aquellos que son adeptos a la filosofía de UNIX y del lenguaje C/C++, el uso de MySQL es muy familiar, ya que su diseño y sus interfaces son acordes a esa filosofía: "crear herramientas que hagan una sola cosa y que la hagan bien".
- MySQL tiene como principal objetivo ser una base de datos fiable y eficiente.

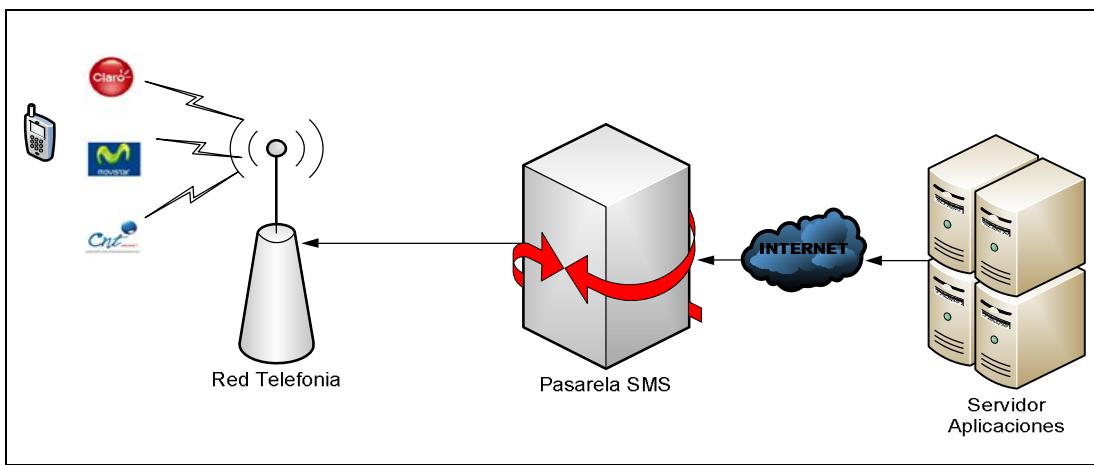
Ninguna característica es implementada en MySQL si antes no se tiene la certeza que funcionará con la mejor velocidad de respuesta y, por supuesto,sin causar problemas de estabilidad.La influencia de C/C++ y UNIX se puede observar de igual manera en su sintaxis. Por ejemplo, la utilización de expresiones regulares, la diferenciación de funciones por los paréntesis, los valores lógicos como 0 y 1, la utilización del tabulador para completar sentencias.

#### **2.5.5 Servicios Móviles (SMS Gateway)**

En la actualidad debido al gran auge que están teniendo los servicios móviles basados en mensajes de texto, existen un amplio número de soluciones confiables y útiles con características destacables; necesarias para enviar mensajes de texto de forma manual o automática desde un ordenador/servidor para prestar servicios de notificaciones, noticias, etc.

Un SMS Gateway (pasarela) es un sistema que permite conectar o enlazar dos sistemas que trabajan con formatos distintos de información, mediante la conversión entre ellos de una forma transparente, permitiendo la comunicación bidireccional entre ellos.

Una pasarela SMS conecta la red de telefonía celular con un conjunto de servicios o aplicaciones SMS, para lo cual generalmente provee de una serie de interfaces para que desarrolladores de software puedan recibir, procesar y enviar un alto número de mensajes; los sistemas que enlaza una pasarela SMS se los puede observar mejor en la Figura 2.7.



**Figura 2.7. Servicios móviles Gateway SMS.**

## 2.6 HERRAMIENTAS DE DESARROLLO

Para el desarrollo del presente proyecto se utiliza varias herramientas relacionadas con cada una de las fases de RUP, tomando como inicio el Análisis y Modelado de la lógica de negocio y la estructura del Sistema (Casos de Uso), Diseño y Modelado de los Datos a utilizar (Modelo Físico y Lógico); hasta llegar a la implementación de la misma (Construcción de Consultas, Carga de Datos y Generación de código).

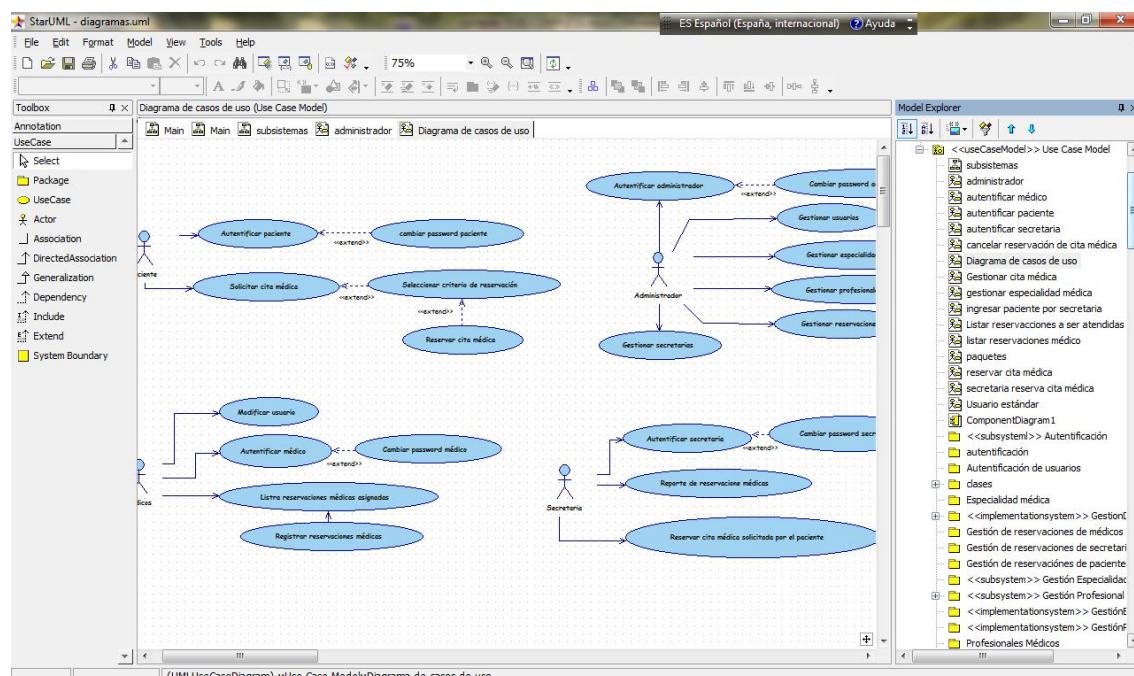
### 2.6.1 STARUML 5.0.2

StarUML 5.0.2 es un completísimo editor gráfico para el desarrollo de sistemas muy necesario en la elaboración y diseño de diagramas UML, permite el diseño de diagramas de clases, componentes, objetos, paquetes, estructuras, módulos, actividades, estados, secuencias, comunicación, interacción, tiempos, etc.

Viene equipado con sus propias plantillas para no tener que diseñar el dibujo a partir de cero y es totalmente compatible tanto con C++ como con Java.

Los diseños y diagramas UML aplicados en este proyecto permiten explicar conceptual y esquemáticamente todo el proceso de la aplicación así como las pautas marcadas para cada uno de los objetivos.

Todos los diagramas UML necesarios para el Análisis y Diseño del presente proyecto son desarrollados en esta herramienta, desde el Diseño del modelo de negocio (**Figura 3.1**); hasta la descripción de los Casos de Uso para cada uno de los actores que intervienen en la aplicación; además de diagramas de Colaboración y Secuencia para detallar de una manera adecuada el funcionamiento del Sistema.



Fuente: Propia

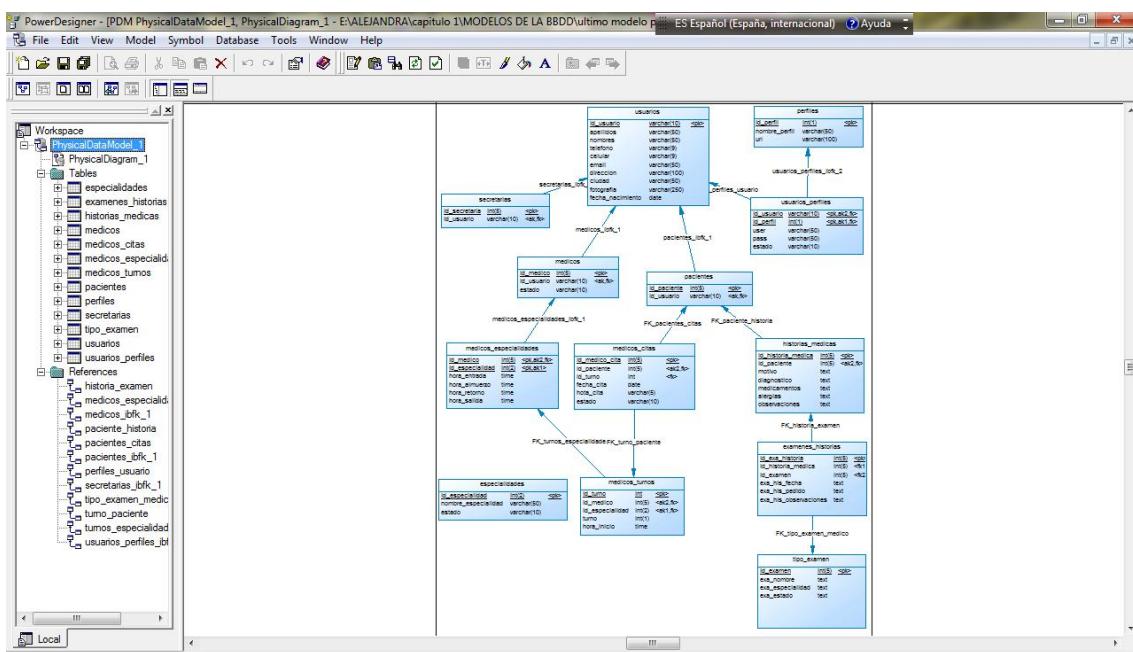
**Figura 2.8.** StarUML 5.0.2

## 2.6.2 Power Designer 11.0

Es una herramienta para el análisis, diseño inteligente y construcción sólida de una base de datos y un desarrollo orientado a modelos de datos a nivel físico y conceptual, que da a los desarrolladores Cliente/Servidor la más firme base para aplicaciones de alto rendimiento.

Ofrece un acercamiento de diseño para optimizar las estructuras de las bases de datos. Capturando el flujo de datos de la aplicación, permite crear un modelo conceptual y físico de la base de datos.

Tanto el Modelo Físico ver (**Figura 3.60**) como Modelo Lógico de la base de Datos fue desarrollada utilizando esta herramienta para detallar de una manera adecuada la relación y manejo de los datos en el funcionamiento del Sistema; así como la generación del script sql necesario para la creación inicial de la Base de Datos del sistema en el Gestor Seleccionado (Mysql).



Fuente: Propia

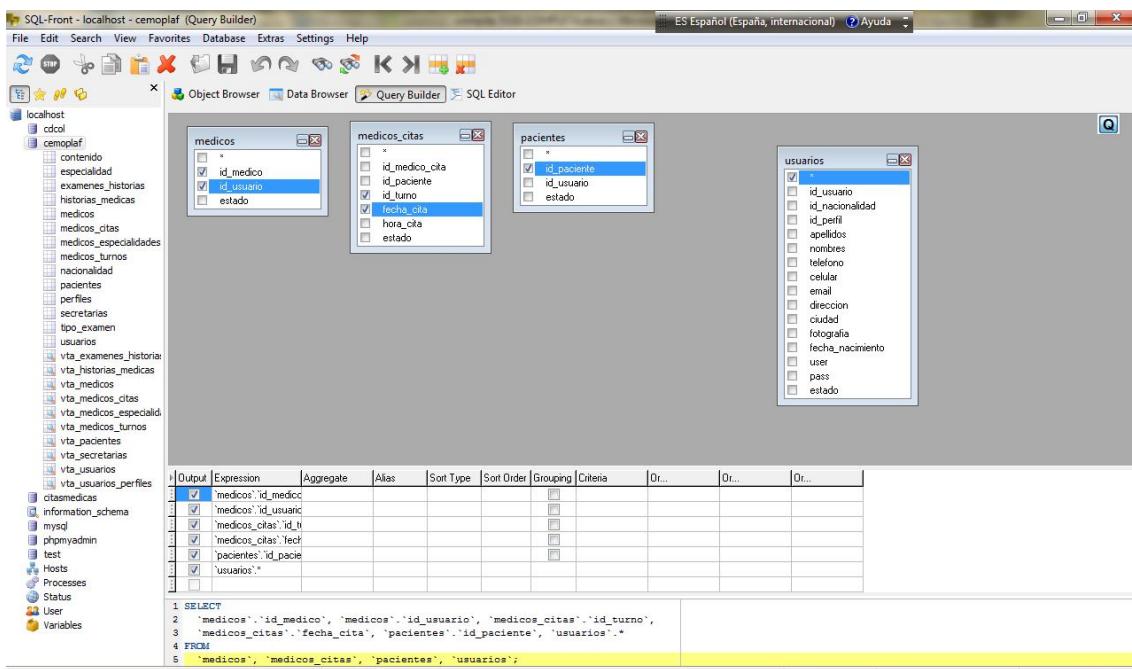
**Figura 2.9.** Power Designer 11.0

### 2.6.3 SQL Front

SQL-Front es una sencilla pero muy útil aplicación diseñada especialmente para desarrollado de aplicaciones que trabajan con Gestores de Bases de Datos MySQL. Con SQL-Front se pueden realizar acciones básicas como añadir, borrar o modificar tablas, campos, registros, entre muchas otras opciones; en un entorno gráfico fácilmente manejable; permite crear muchas consultas de manera visual, seleccionando las tablas involucradas y permitiendo relacionarlas de una manera ágil y sencilla.

Esta herramienta se utiliza dentro del presente proyecto en la relación y manejo de los datos en el funcionamiento del Sistema, carga de datos de prueba, y generación

de las consultas (script) para la obtención de datos necesarios para los reportes y manejo de datos desde el modelo de la aplicación.



Fuente: Propia

**Figura 2.10. SQL-Front**

#### 2.6.4 Zend Studio 9.0

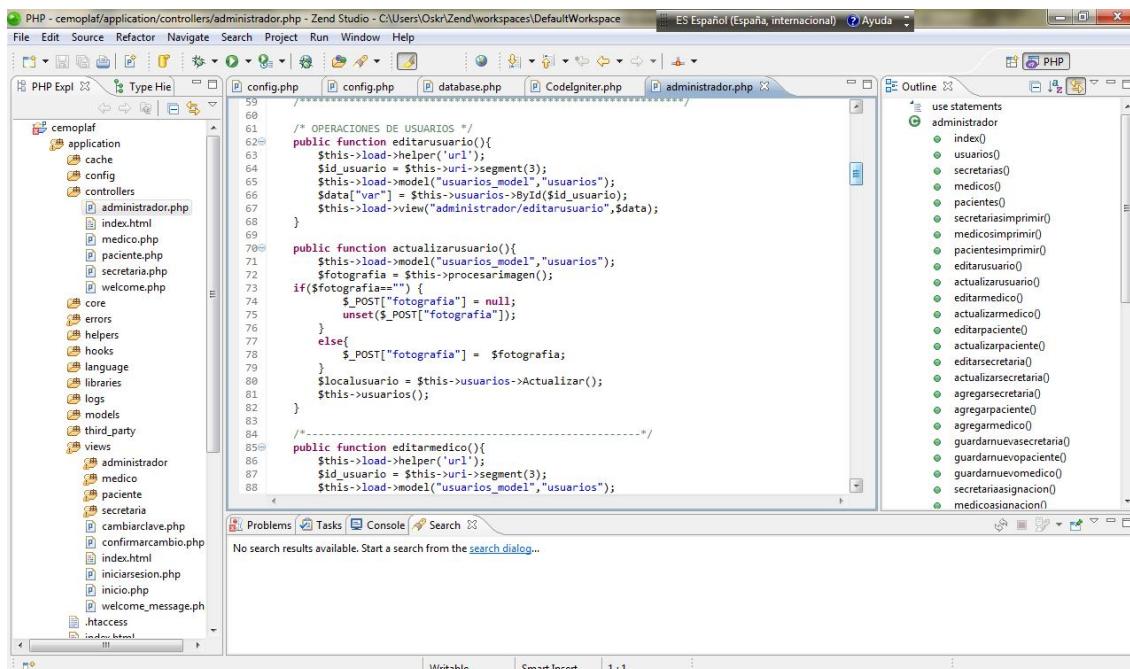
Editor web orientado a la programación de páginas PHP, con ayudas en la gestión de proyectos y depuración de código muy utilizado en proyectos bajo tecnología de servidor PHP, orientada a desarrollar aplicaciones web. El programa, además de servir de editor de texto PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor; permite depuraciones simples de scripts, aunque para la herramienta de depuración es necesario la parte del servidor, que instala Apache y el módulo PHP para trabajar juntos en depuración.

El editor permite escribir los scripts; es bastante útil para la programación en PHP. La interfaz está compuesta por varias partes, en las que encontramos un explorador de archivos, una ventana de depuración, los menús y otra para mostrar el código de las páginas.

Lo contiene una ayuda contextual con todas las librerías de funciones del lenguaje que asiste en todo momento ofreciendo nombres de las funciones y parámetros que deben recibir; permite editar varios archivos y moverse fácilmente entre ellos, marcar a qué elementos corresponden los inicios y cierres de las etiquetas, paréntesis o llaves, moverse al principio o al final de una función.

En la figura 2.11; se muestra el proyecto en la fase de implementación (codificación y generación de páginas), y desarrollo del sistema; el IDE permite mantener una vista tipo árbol de toda la aplicación lo cual es muy conveniente en el desarrollo a través de Codeigniter que es el framework utilizado para la elaboración de esta aplicativo, y tener los tres componentes de la aplicación (Modelo, Controlador, Vista) fácilmente para su exploración y edición.



Fuente: Propia

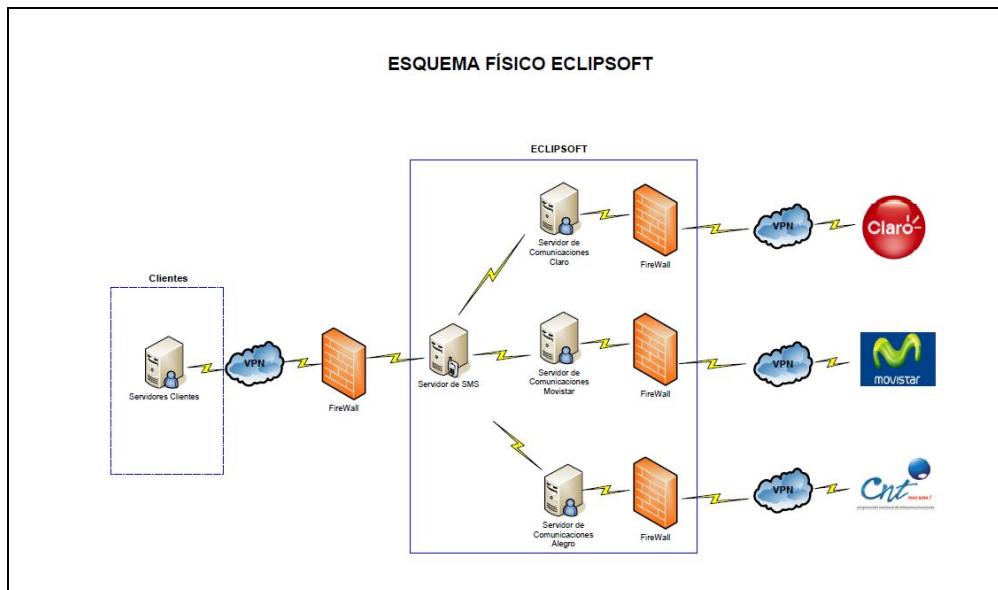
Figura 2.11. Zend Studio 9.0

## 2.6.5 Librería Php Envío SMS

La librería en PHP proporcionada por la empresa Eclipsoft Ecuador, permite agrupar varias funciones y variables en un mismo componente; de manera que luego se pueda incluir esta librería en los proyectos o sitios a través de una cuenta (Usuario y contraseña) registrado anteriormente dentro de un contrato de servicios pactado con la empresa; y disponer de estas funciones fácilmente para realizar envío y recepción de mensajes.

Esta librería debe ser incluida en el proyecto PHP y de esta manera la aplicación programar los envíos, desde sus eventos debidamente programados dentro de la Lógica del funcionamiento del mismo.

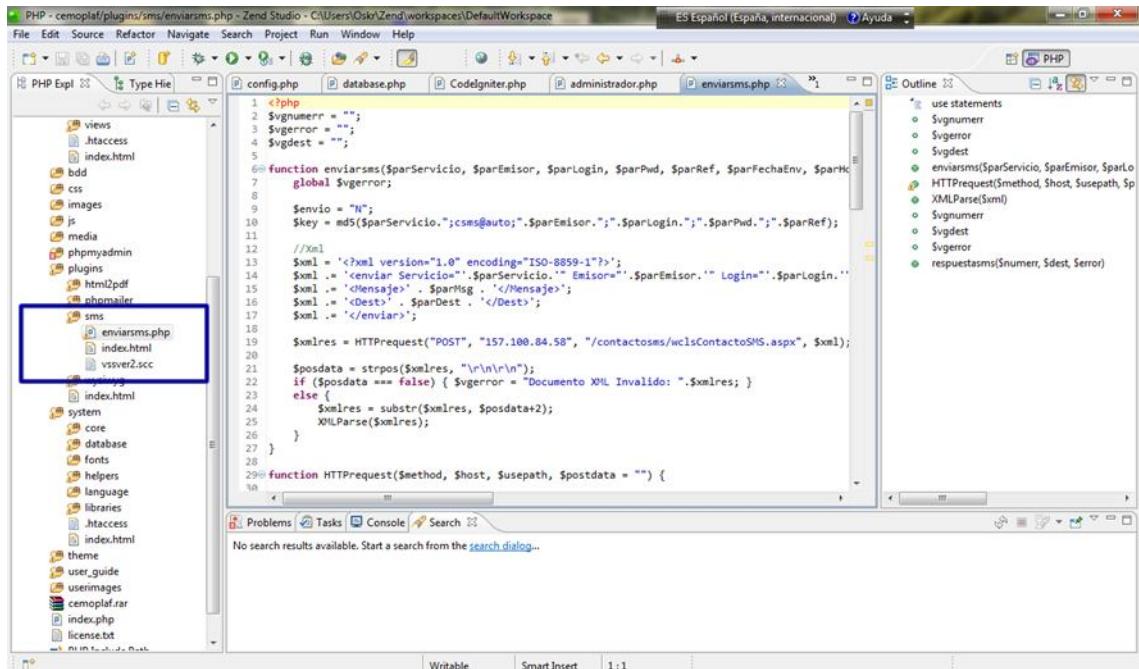
El esquema de funcionamiento se muestra en la siguiente figura de acuerdo a la estructura interna de la empresa proveedora del servicio.



Fuente: ([www15.2012](http://www15.2012))

**Figura 2.12. Esquema Físico Eclipsoft**

La librería necesaria para el envío de SMS desde la aplicación, se muestra en la siguiente figura de acuerdo a la estructura y el lugar necesario dentro de la aplicación (plugin). Esta librería consta de un código proporcionado por la empresa proveedora del servicio de envío y que debe ser incluido en el proyecto en las notificaciones y confirmaciones de las citas médicas como evento del mismo.



The screenshot shows the Zend Studio interface with the following details:

- Title Bar:** PHP - cemoplauf/plugins/sms/enviarsms.php - Zend Studio - C:\Users\Osikr\Zend\workspace\DefaultWorkspace
- Language Bar:** Es Español (España, internacional) / Ayuda
- File Menu:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbars:** Standard, PHP, Outline, Problems, Tasks, Console, Search.
- Left Sidebar:** Shows the project structure with a folder named "sms" highlighted by a blue rectangle. Inside "sms" are files: config.php, database.php, CodeIgniter.php, administrador.php, enviarsms.php, and vsver2.scc.
- Central Editor:** Displays the PHP code for the "enviarsms.php" file. The code uses XML to construct a message and send it via HTTP POST to a specific URL.
- Outline View:** Shows a tree view of the code structure, including nodes like \$vnumer, \$verror, \$vdest, enviarsms(), XMLHttpRequest(), XMLParse(\$xml), \$vnumer, \$vdest, \$verror, and respuestasms().
- Bottom Status Bar:** Shows Writable, Smart Insert, and Line 1:1.

```

1 <?php
2 $vnumer = "";
3 $verror = "";
4 $vdest = "";
5
6 function enviarsms($parServicio, $parEmisor, $parLogin, $parPwd, $parRef, $parFechaEnv, $parHo
7     global $verror;
8
9     $envio = "N";
10    $key = md5($parServicio.".csmss@auto;".$parEmisor.";".$parLogin.";".$parPwd.";".$parRef);
11
12    //Xml
13    $xml = '<?xml version="1.0" encoding="ISO-8859-1"?>';
14    $xml .= '<enviar Servicio="'.$parServicio.'" Emisor="'.$parEmisor.'" Login="'.$parLogin.'"
15    $xml .= '<Mensaje>' . $parMsg . '</Mensaje>';
16    $xml .= '<Dest>' . $parDest . '</Dest>';
17    $xml .= '</enviar>';
18
19    $xmlres = HTTPRequest("POST", "157.100.84.58", "/contactosms/wclsContactoSMS.aspx", $xml);
20
21    $posdata = strpos($xmlres, "\r\n\r\n");
22    if ($posdata === false) { $verror = "Documento XML Invalido: ".$xmlres; }
23    else {
24        $xmlres = substr($xmlres, $posdata+2);
25        XMLParse($xmlres);
26    }
27 }
28
29 function HTTPRequest($method, $host, $usepath, $postdata = "") {
30
31

```

Fuente: Propia

**Figura 2.13.** Librería PHP para el envío de SMS

# CAPÍTULO 3

## DESARROLLO DEL SISTEMA



*Introducción*

*Visión*

*Modelo de Negocios*

*Especificación de Requerimientos*

*Análisis y Diseño*

*Implementación*

*Pruebas al sistema*

*Despliegue*

### **3.1 INTRODUCCIÓN**

Para poder tener una idea clara y entender el funcionamiento del consultorio médico en el cual se implantará LA APLICACIÓN WEB PARA LA ADMINISTRACIÓN ONLINE DE CITAS MÉDICAS, fue necesario realizar visitas periódicas y observar de una manera detenida el trabajo de cada una de las personas que lo integran, para así poder definir cuáles son los procesos que se dan al momento de hacer una reservación de una cita médica.

La información se la obtuvo realizando entrevistas a los actores que integran el consultorio tanto internos(Administrador, Médico) como externos (Pacientes), de esa manera se pudo determinar la secuencia de los procesos, así como las necesidades que existen para solucionarlas mejorando y automatizando dichos procesos.

En primera instancia se estableció la distribución de los médicos dentro del consultorio médico, en la planta baja en servicio al cliente se encuentra una secretaría que es quien realiza las reservaciones de las citas médicas, de acuerdo a los horarios de cada uno de los médicos. Todo este proceso de reservación se lo realiza de una manera manual, lo que contribuye a que se den errores al momento de realizar las reservaciones, como pueden ser cruce de horarios de los pacientes, se puede dar que los médicos no se encuentren presentes en el consultorio para atender las citas, etc.

Para poder optimizar estos procesos de reservación, se establecieron algunas pautas importantes dentro del sistema que las desarrollaremos en este capítulo.

En este capítulo se desarrollan las fases de RUP según su Estructura Dinámica que son: Inicio, Elaboración, Construcción y Transición.

**Fase Inicio:** En esta fase hacemos el análisis de la visión general de los requerimientos del proyecto desde la perspectiva del usuario. Esta fase está sustentada con el documento de visión que es parte de los anexos, (**Anexo 2**).

**Fase Elaboración:** En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura (incluyendo las partes más relevantes y / o críticas del sistema). Al final de esta fase, todos los casos de uso correspondientes serán implementados.

**Fase de Construcción:** Se realiza la implementación, definiendo los subsistemas de implementación, desarrollando los módulos establecidos, implementando el sitio y realizando las pruebas. El fin de esta fase es la finalización del producto, listo para ser entregado.

**Fase de Transición y Despliegue:** En esta fase se asegura la implantación del sistema de manera adecuada, incluyendo el entrenamiento de los usuarios. El fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales de instalación (**Anexo 5**) y todo el material de apoyo al usuario manuales Técnico Funcional (**Anexo 4**), Manual de Usuario, (**Anexo 3**).

## **3.2 VISIÓN**

### **3.2.1 PROPOSITO**

Este documento tiene como propósito recoger, analizar y definir las necesidades de alto nivel y las características del sistema web concerniente al manejo de una APLICACIÓN WEB PARA LA ADMINISTRACIÓN ONLINE DE CITAS MÉDICAS EN EL CENTRO MÉDICO DE ORIENTACIÓN Y PLANIFICACIÓN FAMILIAR CEMOPLAF-OTAVALO.

La Aplicación web deberá estar basada en el cumplimiento de los requisitos previamente establecidos por los Directivos del Centro Médico, la aplicación estará enfocada principalmente en diseñar y desarrollar un sistema que posibilite al Centro Médico de Orientación y Planificación Familiar CEMOPLAF-Otavalo, administrar eficientemente las citas e identificación del historial clínico de sus pacientes.

Este sistema tiene como funcionalidad llevar un control adecuado de todos los procesos concernientes en la administración online de citas médicas.

Algunas funciones principales que debe realizar este sistema son las siguientes:

- Gestionar correctamente las diferentes cuentas de usuario según los perfiles que se establezcan incrementando la seguridad e integridad de la información que maneja la aplicación.
- Permitir al Médico administrar la identificación de historiales clínicos de sus pacientes y obtener reportes instantáneos de sus citas realizadas, suspendidas, pendientes o eliminadas; independientemente del lugar donde se encuentre.
- Permitir al paciente hacer una cita online al centro médico así como su identificación de historial clínico de forma personalizada.

### **3.2.2 ALCANCE**

El alcance del “Aplicación Web para la Administración Online de Citas Médicas en el Centro Médico de Orientación y Planificación Familiar CEMOPLAF-Otavalo” cubrirá todo el control de citas del centro médico, el cual está dirigido a todos los pacientes.

Este documento especifica el problema u oportunidad a resolver, los involucrados y usuarios afectados, y la solución ofrecida por la Aplicación Web para la Administración Online de Citas Médicas en el Centro Médico de Orientación y Planificación Familiar CEMOPLAF-Otavalo. Describe la funcionalidad y características de la solución en general. Los detalles sobre la funcionalidad del sistema se encuentran en el Modelo de Casos de Uso.

### **3.2.3 OPORTUNIDAD DE NEGOCIO**

Mediante la realización de este sistema web se podría agilizar el manejo de las citas médicas con el cual se ahorraría tiempo y se cumpliría los requerimientos de los pacientes y así se mejoraría el servicio.

Permite al centro contar con un registro todas las citas realizadas en las diferentes especialidades, como también un registro de los pacientes. Ofrecerá un control eficiente de las citas a través de las respectivas notificaciones a los médicos y gracias a interfaces amigables y sencillas. Con el tiempo esto conllevara a tener más usuarios o pacientes que accedan a los servicios que presta el centro médico, habrá más oportunidad de trabajo para médicos, secretarias, administradores y personas que colaboran en la institución.

<b>El problema</b>	Una de las debilidades es la atención externa que CEMOPLAF Otavalo brinda a sus usuarios, en la que la solicitud de turno para ser atendido se lo debe realizar de manera personal en las instalaciones una hora antes de que médico llegue al consultorio, lo que implica que el usuario tiene que hacer largas filas para reservar su cita médica y luego regresar horas más tarde para poder acceder a la consulta de acuerdo al número de pacientes que el médico tenga asignado o esperar en las instalaciones a ser atendido, es por ello que las salas de espera siempre están llenas de usuarios; otra opción es solicitar su turno a través de una llamada telefónica que conlleva a mantener ocupada la línea del centro hasta concluir con la asignación de su pedido; siendo a veces imposible poder comunicarse.
<b>Afecta a</b>	Al centro porque tiene deficiencia en la realización de las citas médicas. A los pacientes porque el tiempo de espera para realizar una cita demora.
<b>El impacto asociado es</b>	<ul style="list-style-type: none"> <li>• Los procesos no óptimos que realiza el centro.</li> <li>• Perdida de pacientes.</li> <li>• Provoca pérdida de tiempo tanto para pacientes como para doctores.</li> </ul>
<b>Una adecuada solución sería</b>	Implementar una Aplicación Web para la Administración Online de Citas Médicas en el Centro Médico de Orientación y Planificación Familiar CEMOPLAF-Otavalo.

Fuente: Propia

**Tabla 3.1.** Sentencias que definen el problema

<b>Para</b>	<b>Para los diferentes pacientes.</b>
<b>Quienes</b>	Necesitan sistemas confiables para la Administración Online de Citas Médicas.
<b>El nombre del producto</b>	CEMOCITE, Versión 1.1
<b>Que</b>	Permitirá obtener beneficios como: comodidad y rapidez. A través de Internet, en tiempo real, y eligiendo el día y la hora más conveniente con el médico idóneo, modificarla o incluso, llegado el caso, anularla. Y todo esto desde cualquier ordenador. Al mismo tiempo, nuestro sistema, proporciona un control exacto de las citas a través de una Agenda, y en el caso de Servicios de Salud, le proporciona al médico la posibilidad de mantenerse en contacto con su paciente a través e-mail u otras alternativas de comunicación en línea, estableciendo un vínculo entre pacientes y profesionales médicos.
<b>Nuestro producto</b>	Será escalable, distribuido, extensible, minimizará los problemas del centro, y permitirá almacenar información necesaria acerca de las citas médicas, de esta manera brindar un mejor servicio a los pacientes. Además proporciona un acceso rápido y actualizado a la información desde cualquier punto dentro del alcance geográfico del sistema web.

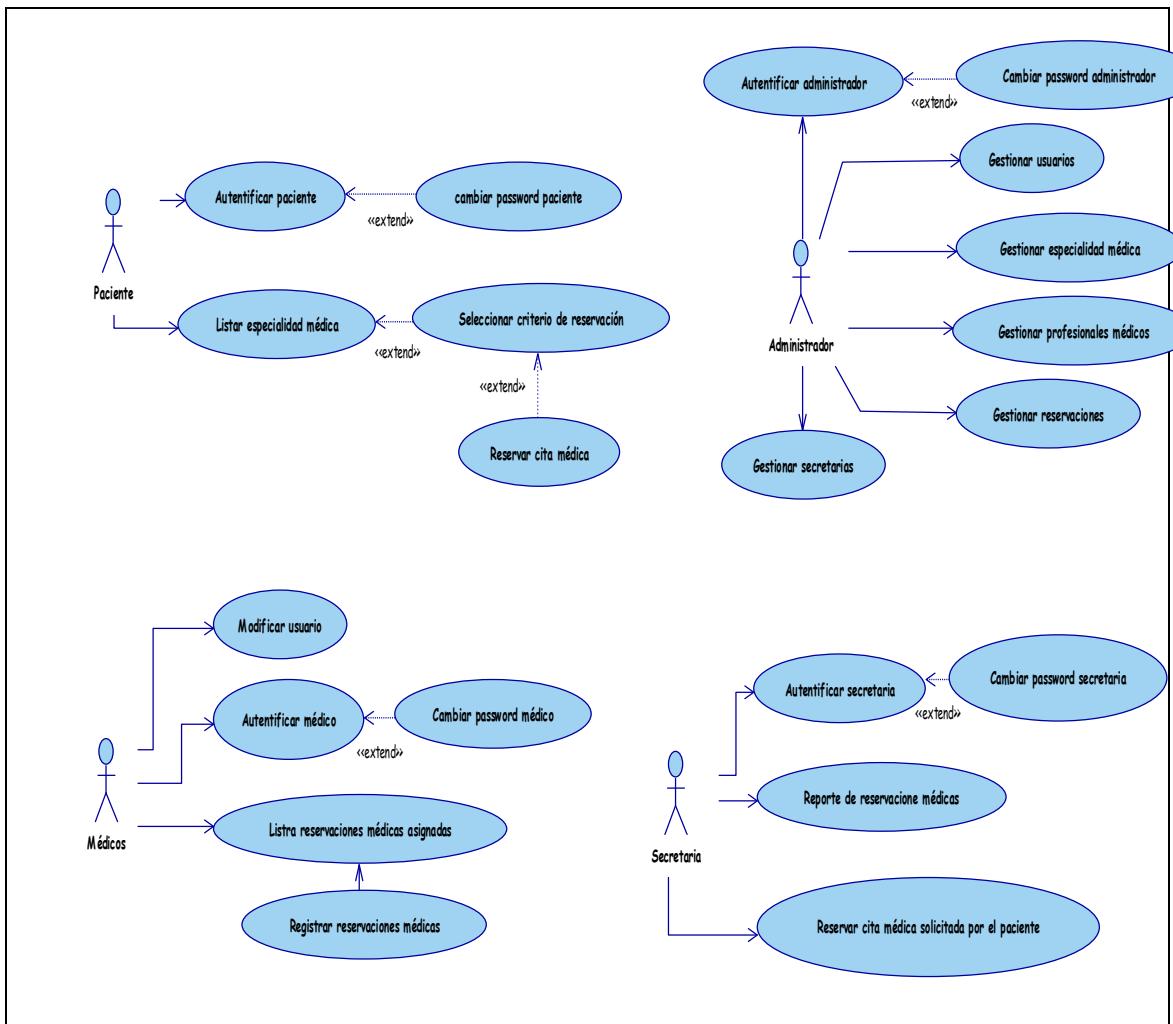
Fuente: Propia

**Tabla 3.2.** Sentencias que definen la posición del Producto

### 3.3 MODELO DE NEGOCIOS

La Aplicación Web Para La Administración Online De Citas Médicas En El Centro Médico De Orientación Y Planificación Familiar CEMOPLAF-OTAVALO, permitirá automatizar la reserva de citas médicas, lo que supondrá un acceso rápido y sencillo a la base de datos, gracias a interfaces gráficas y amigables, además los datos accedidos están siempre actualizados, lo cual es muy importante para obtener

la información en tiempo real, desde el lugar donde se encuentre las 24 horas del día y los 365 días al año.



Fuente: Propia

**Figura 3.1.** Diagrama del Modelo de Negocios CEMOCITE

### 3.4 ESPECIFICACIONES DE REQUERIMIENTOS

Los requerimientos son la pieza fundamental de un proyecto de desarrollo de software, describiendo técnicamente las funciones, el desempeño y las características que el producto debe tener, dentro de estas especificaciones para el desarrollo del proyecto se toma en cuenta los siguientes pasos:

- Planear el proyecto y los recursos que se usarán en este.
- Especificar el tipo de verificaciones que se habrán de realizar al sistema.
- Planear la estrategia de prueba a la que habrá de ser sometido el sistema.

El presente proyecto propone un sistema que sea capaz de permitir la LA ADMINISTRACIÓN ONLINE DE CITAS MÉDICAS EN EL CENTRO MÉDICO DE ORIENTACIÓN Y PLANIFICACIÓN FAMILIAR CEMOPLAF-OTAVALO; publicación de horarios, reservación de citas médicas, además de la consulta de las reservaciones médicas que deben ser atendidas por el profesional médico, mediante el uso de Internet. La aplicación tendrá como nombre CEMOCITE.

### **3.4.1 Requerimientos de Entorno**

Los requerimientos de entorno son aquellos que rodean al sistema, lo usan y lo necesitan como una fuente de los servicios necesarios para que funcione. El sistema deberá ser implementado en un servidor de aplicaciones web disponible las 24 horas del día todos los días del año, con los siguientes requisitos:

- Gestor de base de datos
- Interprete de PHP versión 5 o superior
- Sistema operativo compatible con Java script
- Conexión a internet con tasa de transferencia aceptable igual o superior a 64kbps
- Visualización de todos los componentes del sitio en los diferentes navegadores (mozilla, internet explorer, opera y chrome).

### **3.4.2 Requerimientos funcionales**

Los requerimientos funcionales son aquellos que describen lo que el sistema debe de hacer. Estos requerimientos al tiempo que avanza el proyecto de software se convierten en los algoritmos, la lógica y gran parte del código del sistema.

El sistema tendrá dos partes funcionales, una parte se ejecuta por parte del usuario cuando accede al internet para realizar la reservación de la cita médica y la otra parte es la administrativa, alojada en el servidor de aplicaciones web.

#### **Requerimientos funcionales para la administracion**

Los requerimientos funcionales para la parte administrativa del sistema son:

- **Autentificación de usuarios:** Se refiere a los perfiles de usuarios, El sistema contará con los siguientes perfiles de usuario:
  - Administrador
  - Paciente
  - Médico

- Secretaria

Para el ingreso al sitio Web CEMOCITE se tendrá que ingresar un login y un password para cada usuario de acuerdo a su perfil.

- **Administración de especialidades medicas:** Se entiende por el ingreso, modificación, visualización y actualización de las diferentes especialidades que ofrece el centro médico.
- **Administración de usuarios finales:** Se entiende por administración de usuarios al registro, actualización y modificación de los datos de los usuarios finales.
- **Administración de profesionales médicos:** Se entiende por administración de profesionales médicos al registro, actualización y modificación de los datos de los profesionales médicos. El registro y modificación de los profesionales médicos se los realizará desde el sistema administrador CEMOCITE. El centro médico será el encargado de la modificación y asignación de horarios al profesional médico, por medio del sitio Web CEMOCITE.
- **Administración de reservación de citas medicas:** Desde el sistema CEMOCITE se podrá visualizar las reservaciones realizadas por el usuario final.
- **Administración de horarios:** El centro médico será el encargado de fijar horarios para la atención médica de acuerdo a las especialidades disponibles.
- **Vizualización de reservaciones:** El centro médico dispondrá del sitio Web para la vizualización de reservaciones realizadas por fechas.

#### **Requerimientos funcionales para los dispositivos del usuario**

Los requerimientos funcionales para los dispositivos del usuario final (dispositivos móviles, pcs) que permite el uso de la aplicación desde un dispositivo son:

- **Autentificación de usuarios:** Al ingresar al sistema, el usuario debe autenticarse, de acuerdo a los perfiles definidos:
  - El administrador que es el que maneja el sistema y tiene todos los permisos para manejar el sistema.
  - Usuario final, comprende todo tipo de usuarios que ingresen al sitio web, tendrá acceso para realizar solicitud de citas médicas, revisión de datos médicos, Información de médicos disponibles de acuerdo a los horarios de atención.
  - Profesional médico, perfil que le permite revisar las consultas que tiene que atender de acuerdo al horario de atención, tener un reporte de las citas pendientes, cancelación de citas, notificaciones importantes (vía correo y SMS).

- La secretaria que será la persona que mediante llamadas telefónicas podrá reservar turnos en el sitio web.
- **Publicación de horarios de atención:** El usuario puede consultar desde su dispositivo los horarios de atención médica disponibles de todas las especialidades y servicios que ofrece el centro médico.
- **Reservación de citas medicas:** Una vez que el usuario final ha seleccionado el horario de su preferencia, realiza la reservación por el mismo medio. Esta función se efectuará siempre y cuando se encuentre dentro de la cobertura.
- **Reportes de reservación de citas:** El profesional médico puede obtener el reporte de las reservaciones que deben ser atendidas.
- **Reportes de citas concretadas:** La secretaria puede tener un reporte de las citas realizadas por el médico.

#### **3.4.3 Requerimientos no funcionales**

La Aplicación Web para la Administración Online de Citas Médicas en el Centro Médico de Orientación y Planificación Familiar CEMOPLAF-Otavalo, contara con una interfaz amigable, dinámica es decir con un menú y sus respectivas ayudas para facilitar la navegación en el sistema y así los usuarios podrán registrarse y solicitar una citas con el médico; de igual manera en los colores del sistema que son de acuerdo al centro al médico en este caso es el color azul y blanco.

#### **3.4.4 Descripción de actores**

Los actores son aquellos que interactúan con el sistema. Representan a todos los que necesita intercambiar información con el sistema.

Los actores que interactúan con CEMOCITE, se listan en la siguiente tabla:

ACTOR	DESCRIPCION	PERFIL
<b>Administrador</b>	El Administrador es el único que puede crear a los médicos y secretarias ya que esta es una tarea propiamente administrativa; a la vez que también podrá crear pacientes.	El perfil de usuario Administrador es el único que podrá realizar todas las tareas que posee el sistema. Sera una persona capacitada que conozca el sistema.
<b>Médico</b>	Los médicos establecerán sus horarios de atención en el centro médico; pero dicho ingreso de horarios los puede realizar el administrador, la secretaria o el respectivo médico	Doctor especializado en cualquier campo de la medicina, con conocimientos del manejo de sistema.
<b>Secretaria</b>	La secretaria puede realizar una reservación de un cita médica en el sistema solicitada por un paciente vía telefónica.	Cualquier persona que conosca el uso del sistema y pueda hacer uso de este.
<b>Paciente</b>	El paciente se registra directamente desde la Web llenando un formulario, no tendrá que realizar su registro en el Centro médico Cemoplaf.	Cualquier persona que necesite acceder a los servicios que presta el centro médico, y que tenga la capacidad de ingresar al sitio web.

Fuente: Propia

**Tabla 3.3.** Descripción de actores

Los perfiles de usuario Paciente, Médico o Secretaria podrán realizar determinadas tareas del sistema las cuales se le vera con más claridad en los Diagramas de Casos de uso.

Cada uno de los usuarios de los perfiles antes mencionados deberán encontrarse activos, caso contrario no se les permitirá ingresar a sus respectivos módulos.

Los usuarios de cada uno de los perfiles del sistema tendrán que logearse para ingresar al mismo, el sistema de reservaciones online verificará el tipo de perfil de usuario, además comprobará si se encuentra activo, una vez establecido esto le dará ingreso al módulo que le corresponda a este perfil de usuario; caso contrario no le permitirá ingresar a dicho módulo y le regresara a la página de logeo.

El sistema verificará los horarios de atención de los médicos, y la disponibilidad de los mismos, y aceptará o rechazará la reservación de la cita médica por parte del paciente, o de la secretaria.

La reservación de la cita médica una vez aceptada por el sistema tomara un estado de registrada, esperando a que se establezca lo que cambiara de estado a ejecutada.

Las reservaciones de las citas médica también pueden ser canceladas por los pacientes o los médicos siempre y cuando la notificación no haya sido enviada, y así a la reservación de la cita se le pondrá el estado de cancelada ya sea por el paciente o por la secretaria.

Después de haber establecido estas pautas para el sistema se desarrollaron los respectivos Diagramas de Casos de Uso para poder tener una idea más clara y visual de cómo son los procesos y a lo que se enfocará el sistema de reservaciones online.

### **3.4.5 Descripción de casos de uso**

Un caso de uso representa una unidad funcional coherente del sistema. Los casos de uso son parte del análisis ya que ayudan a describir qué es lo que el sistema debe hacer, desde el punto de vista del usuario. Es decir, describen un uso del sistema y cómo este interactúa con el usuario.

Los actores identificados para el desarrollo de los Diagramas de Casos de Uso son los siguientes:

- Administrador
- Paciente
- Médico
- Secretaria

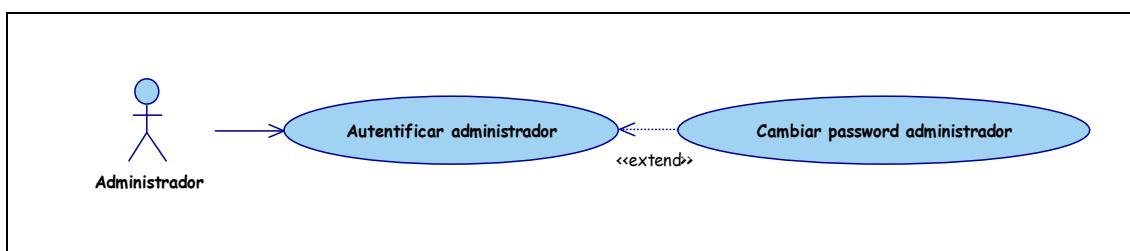
Los diagramas de casos de uso identificados son los siguientes:

#### **CASOS DE USO DEL ACTOR: ADMINISTRADOR**

Los casos de uso se usan para especificar el comportamiento del sistema

CEMOCITE en el módulo de administración son:

- **Caso de uso Autentificar usuario Administrador**



Fuente: Propia

**Figura 3.2.** Caso de uso Autentificar usuario administrador

Caso de uso:	Autenticar usuario administrador
Descripción:	Permite ingresar a CEMOCITE mediante un login y un password.
Actores:	Administrador
Acciones básicas:	<ol style="list-style-type: none"> <li>Conectar al sistema CEMOCITE.</li> <li>Ingresar login y password.</li> </ol>
Acciones alternativas:	Si el login o password no son válidos, el sistema muestra un mensaje de error.
Post-Condición:	El sistema permite el acceso al sistema de administración.

Fuente: Propia

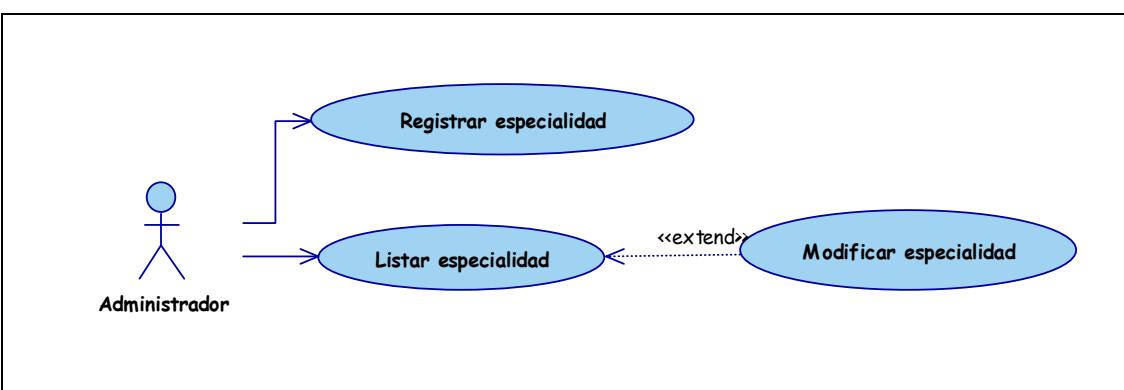
**Tabla 3.4.** Caso de Uso Módulo Administración: Autenticar usuario Administrador

Caso de uso:	Cambiar password administrador
Descripción:	Permite cambiar el password del administrador.
Actores:	Administrador
Acciones básicas:	<ol style="list-style-type: none"> <li>Conectar al sistema CEMOCITE.</li> <li>Cambiar password</li> </ol>
Acciones alternativas:	Si el login o password no son válidos, el sistema muestra un mensaje de error.
Post-Condición:	El sistema procesa el cambio de password y Permite acceso al sistema de administración.

Fuente: Propia

**Tabla 3.5.** Caso de Uso Módulo Administración: Cambiar password administrador.

- Caso de uso Gestionar Especialidad Médica**



Fuente: Propia

**Figura 3.3.** Caso de uso Gestionar Especialidad Médica

Caso de uso:	Registrar especialidad médica
Descripción:	Permite ingresar una nueva especialidad médica al sistema.
Actores:	Administrador
Pre-Condiciones:	El perfil del usuario debe ser administrador.
Acciones básicas:	<ol style="list-style-type: none"> <li>Ingresar al módulo de las especialidades médicas.</li> <li>Ingresar los datos correspondientes a la nueva especialidad médica.</li> <li>Guardar información.</li> </ol>
Acciones alternativas:	El sistema validará si la especialidad ingresada existe o no.
Post-Condición:	El sistema debe permitir la gestión de la nueva especialidad.

Fuente: Propia

**Tabla 3.6.** Caso de Uso Módulo Administración: Registrar Especialidad médica

Caso de uso:	Listar especialidad médica
Descripción:	Permite desplegar una lista de las especialidades médicas existentes en el sistema.
Actores:	Administrador
Pre-Condiciones:	Ingreso al sistema con cuenta usuario administrador.
Acciones básicas:	<ol style="list-style-type: none"> <li>Ingresar al módulo de especialidades médicas.</li> <li>Listar especialidades médicas.</li> </ol>
Post-Condición:	Visualizar las especialidades médicas.

Fuente: Propia

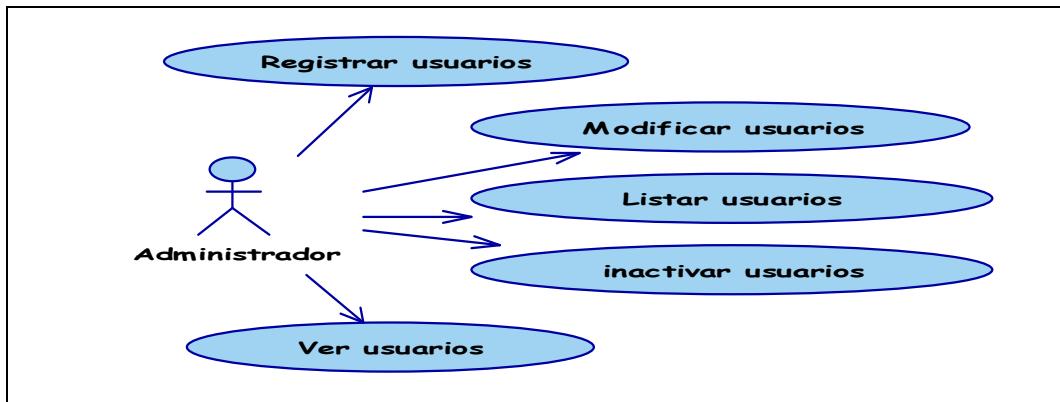
**Tabla 3.7.** Caso de Uso Módulo Administración: Listar especialidad médica.

Caso de uso:	Modificar especialidad médica
Descripción:	Permite cambiar el estado de la especialidad médica, que puede ser activo o inactivo
Actores:	Administrador
Pre-Condiciones:	La especialidad debe existir en el sistema.
Acciones básicas:	<ol style="list-style-type: none"> <li>Listar las especialidades médicas existentes.</li> <li>Desplegar la especialidad médica seleccionada.</li> <li>Modificar el estado de la especialidad médica.</li> </ol> <p>Guardar información.</p>
Acciones alternativas:	El sistema comprueba la validez de los datos, si los datos no son correctos se avisa al actor de ello permitiéndole su corrección.
Post-Condición:	El sistema permite ingresar al registro modificado con los datos actualizados.

Fuente: Propia

**Tabla 3.8.** Caso de Uso Módulo Administración: Modificar especialidad médica

- **Caso de uso Gestionar Usuarios: paciente, médico, secretaria**



Fuente: Propia

**Figura 3.4.** Caso de uso Gestionar usuarios (paciente, médico secretaria)

Caso de uso:	Registrar usuarios
Descripción:	Permite el registro de los usuarios por medio del sitio Web.
Actores:	Administrador
Pre-Condiciones:	El administrador tiene los datos de los usuarios para registrar en la aplicación web.
Acciones básicas:	<ol style="list-style-type: none"> <li>1. Conexión con el sitio Web CEMOCITE.</li> <li>2. Ingresar los datos necesarios del usuario.</li> <li>3. Guarda la información ingresada.</li> </ol>
Acciones alternativas:	Si uno de los datos no son válidos, el sistema muestra un mensaje de error.
Post-Condición:	CEMOCITE devuelve la confirmación del registro creado permitiendo el acceso con su login y password registrados.

Fuente: Propia

**Tabla 3.9.** Caso de Uso Módulo Administración: Registrar usuarios

Caso de uso:	Modificar usuarios
Descripción:	Permite la actualización de datos de los usuarios
Actores:	Administrador
Pre-Condiciones:	El usuario debe estar registrado.
Acciones básicas:	<ol style="list-style-type: none"> <li>1. Cambiar los datos requeridos.</li> <li>2. Guardar la información ingresada.</li> </ol>
Acciones alternativas:	El sistema comprueba la validez de los datos, si los datos no son correctos se avisa al actor de ello permitiéndole que los corrija.
Post-Condición:	CEMOCITE debe actualizar los datos del usuario.

Fuente: Propia

**Tabla 3.10.** Caso de Uso Módulo Administración: Modificar usuarios

Caso de uso:	Listar usuarios
Descripción:	Se despliega la lista de usuarios existentes en el sistema administrador CEMOCITE.
Actores:	Administrador
Pre-Condiciones:	Ingreso al sistema con cuenta usuario administrador.
Acciones básicas:	<ol style="list-style-type: none"> <li>1. Ingresar al módulo correspondiente.</li> <li>2. Listar usuarios por perfil seleccionado.</li> </ol>
Post-Condición:	Realizar cualquier operación necesaria sobre uno de los usuarios.

Fuente: Propia

**Tabla 3.11.** Caso de Uso Módulo Administración: Listar usuarios

Caso de uso:	Ver usuarios
Descripción:	Permite visualizar los datos de los usuarios.
Actores:	Administrador
Pre-Condiciones:	Ingreso al sistema con cuenta usuario administrador.
Acciones básicas:	<ol style="list-style-type: none"> <li>1. Listar usuarios.</li> <li>2. Ver la información correspondiente al usuario requerido.</li> </ol>
Post-Condición:	El sistema presenta la información del usuario.

Fuente: Propia

**Tabla 3.12.** Caso de Uso Módulo Administración: Ver usuarios

Caso de uso:	Inactivar usuarios
Descripción:	Inactivar los usuarios que no han usado la aplicación
Actores:	Administrador
Pre-Condiciones:	Los usuarios inactivados no deben haber usado dentro de un tiempo determinado la aplicación.
Acciones básicas:	<ol style="list-style-type: none"> <li>1. Ingresar al módulo de usuarios.</li> <li>2. Listar los usuarios inactivos por perfil</li> </ol>
Post-Condición:	El sistema administrador CEMOCITE actualiza la lista de los usuarios por perfil seleccionado.

Fuente: Propia

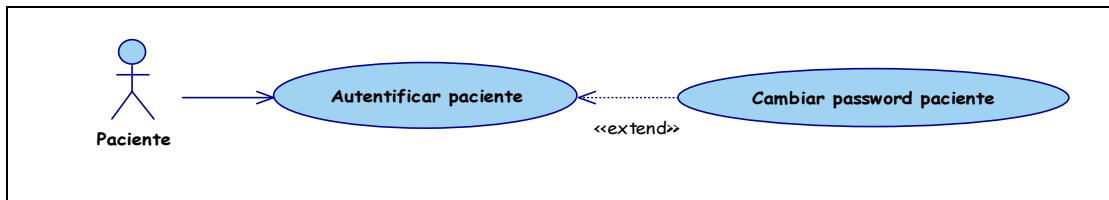
**Tabla 3.13.** Caso de Uso Módulo Administración: Inactivar usuarios

## CASOS DE USO DEL ACTOR: PACIENTE

Los casos de uso se usan para especificar el comportamiento del sistema

CEMOCITE con el paciente son:

- **Caso de uso Autentificar Paciente**



Fuente: Propia

**Figura 3.5.** Caso de uso Autentificar paciente.

Caso de uso:	Autentificar paciente
Descripción:	Permite ingresar a CEMOCITE mediante un login y un password desde el dispositivo.
Actores:	Paciente
Pre-Condiciones:	El paciente ya debe estar registrado en el sitio Web CEMOCITE.
Acciones básicas:	<ol style="list-style-type: none"> <li>1. Conexión con CEMOCITE desde el dispositivo.</li> <li>2. Ingresar login y password.</li> </ol>
Acciones alternativas:	Si el login o password no son válidos, muestra un mensaje de error.
Post-Condición:	El sistema permite el acceso a la aplicación.

Fuente: Propia

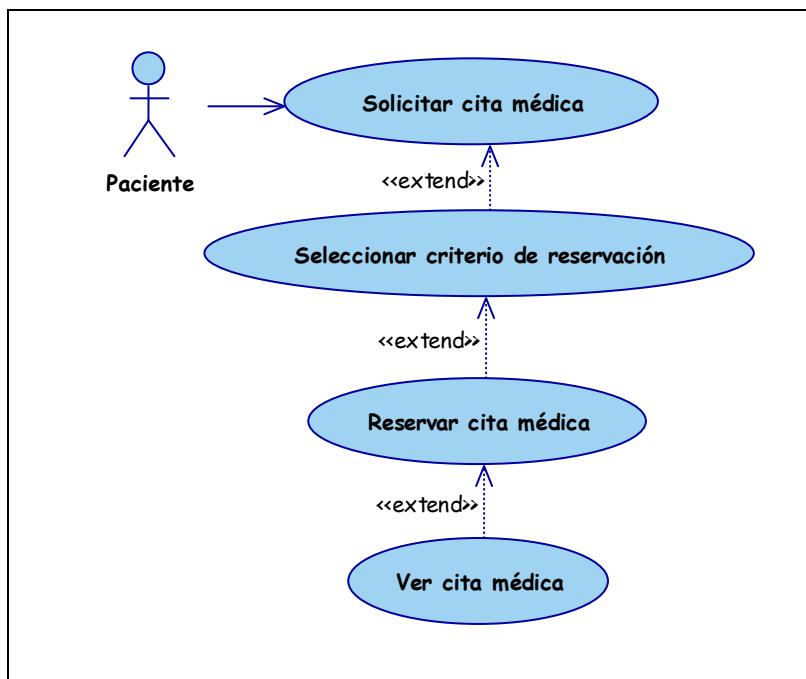
**Tabla 3.14.** Caso de Uso Módulo usuarios: Autentificar paciente.

Caso de uso:	Cambiar password paciente
Descripción:	Permite cambiar el password del paciente.
Actores:	Paciente
Acciones básicas:	<ol style="list-style-type: none"> <li>1. Conectar al sistema CEMOCITE.</li> <li>2. Cambiar password</li> </ol>
Acciones alternativas:	Si el login o password no son válidos, el sistema muestra un mensaje de error.
Post-Condición:	El sistema procesa el cambio de password y Permite acceso al sistema

Fuente: Propia

**Tabla 3.15.** Caso de Uso Módulo usuarios: Cambiar password paciente.

- Caso de uso Reservación de citas médicas



Fuente: Propia

**Figura 3.6.** Caso de uso Reservación de citas médicas.

Caso de uso:	Reservación de cita médica
Descripción:	Permite realizar reservaciones de citas médicas desde la aplicación a través de un dispositivo.
Actores:	Paciente
Pre-Condiciones:	El paciente tiene que estar registrado
Acciones básicas:	Seleccionar criterio de reservación <ol style="list-style-type: none"> <li>1. Seleccionar especialidad</li> <li>2. Seleccionar médico</li> <li>3. Seleccionar horario</li> </ol>
Post-Condición:	CEMOCITE debe responder un mensaje de reservación satisfactoria. Y enviar un correo electrónico de confirmación y SMS.

Fuente: Propia

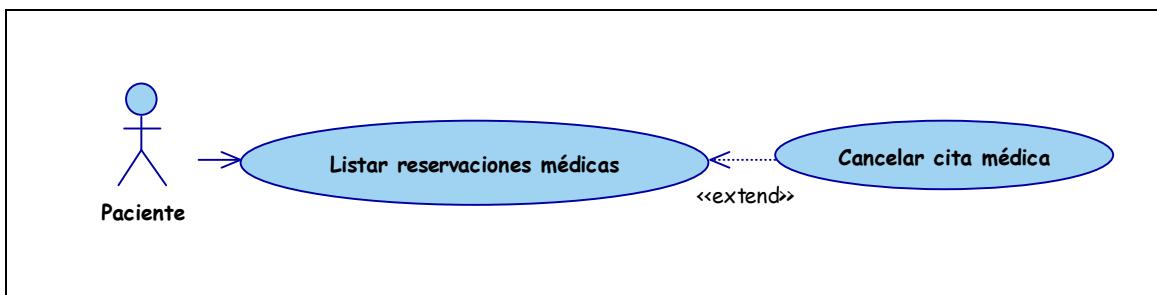
**Tabla 3.16.** Caso de Uso Módulo usuarios: Reservación de cita médica

Caso de uso:	Ver reservación de citas médicas
Descripción:	Permite ver la información de una cita médica seleccionada.
Actores:	Paciente
Acciones básicas:	1. Listar citas médica <ol style="list-style-type: none"> <li>2. Ver información de una cita médica específica</li> </ol>
Post-Condición:	El sistema permite el acceso a la información.

Fuente: Propia

**Tabla 3.17.** Caso de Uso Módulo usuarios: Ver reservación de citas médicas

- Caso de uso Cancelar reservación de citas médicas



Fuente: Propia

**Figura 3.7** Caso de uso Cancelar reservación de citas médicas

Caso de uso:	Cancelar reservación de citas médicas
Descripción:	Permite realizar la cancelación de la cita médica realizada.
Actores:	Paciente
Pre-Condiciones:	El paciente debe estar registrado en el sistema CEMOCITE. El usuario debe haber realizado previamente una reservación.
Pre-Condición:	En este caso se puede eliminar la cita si todavía no ha sido confirmada por el centro médico, ya que el centro médico confirma la cita al siguiente día
Acciones básicas:	<ol style="list-style-type: none"> <li>1. Ingresar a la aplicación desde un dispositivo conectado a internet.</li> <li>2. Listar reservaciones de citas médicas.</li> <li>3. Seleccionar la reservación.</li> <li>4. Cancelar la reservación.</li> </ol>
Post-Condición:	CEMOCITE debe actualizar la lista de reservaciones realizadas por el paciente.

Fuente: Propia

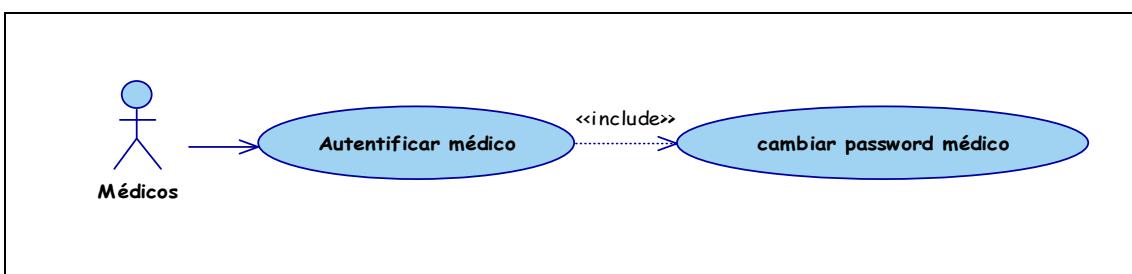
**Tabla 3.18.** Caso de Uso Módulo usuarios: Cancelar reservación de cita médica

### CASOS DE USO DEL ACTOR: MÉDICO

Los casos de uso se usan para especificar el comportamiento del sistema

CEMOCITE con el médico son:

- Caso de uso Autenticar profesional médico



Fuente: Propia

**Figura 3.8.** Caso de uso Autenticar profesional médico

Caso de uso:	Autenticar médico
Descripción:	Autenticar profesional médico
Actores:	Permite ingresar a CEMOCITE mediante un login y un password desde el dispositivo.
Pre-Condiciones:	Profesional médico
Acciones básicas:	El usuario debe estar registrado en el sistema CEMOCITE.
Acciones alternativas:	Si el login o password no son válidos, muestra un mensaje de error.
Post-Condición:	El sistema permite el acceso a la aplicación.

Fuente: Propia

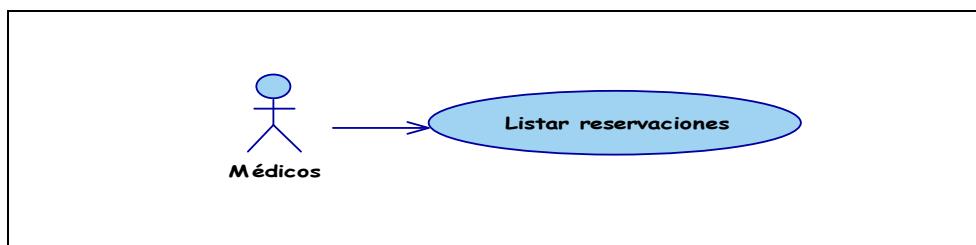
**Tabla 3.19.** Caso de Uso Módulo usuarios: Autenticar profesional médico.

Caso de uso:	Cambiar password médico
Descripción:	Permite cambiar el password del médico
Actores:	Médico
Acciones básicas:	1. Conectar al sistema CEMOCITE. 2. Cambiar password
Acciones alternativas:	Si el login o password no son válidos, el sistema muestra un mensaje de error.
Post-Condición:	El sistema procesa el cambio de password y Permite acceso al sistema

Fuente: Propia

**Tabla 3.20.** Caso de Uso Módulo usuarios: Cambiar password médico.

- **Caso de uso Listar reservaciones que deben ser atendidas por el profesional médico**



Fuente: Propia

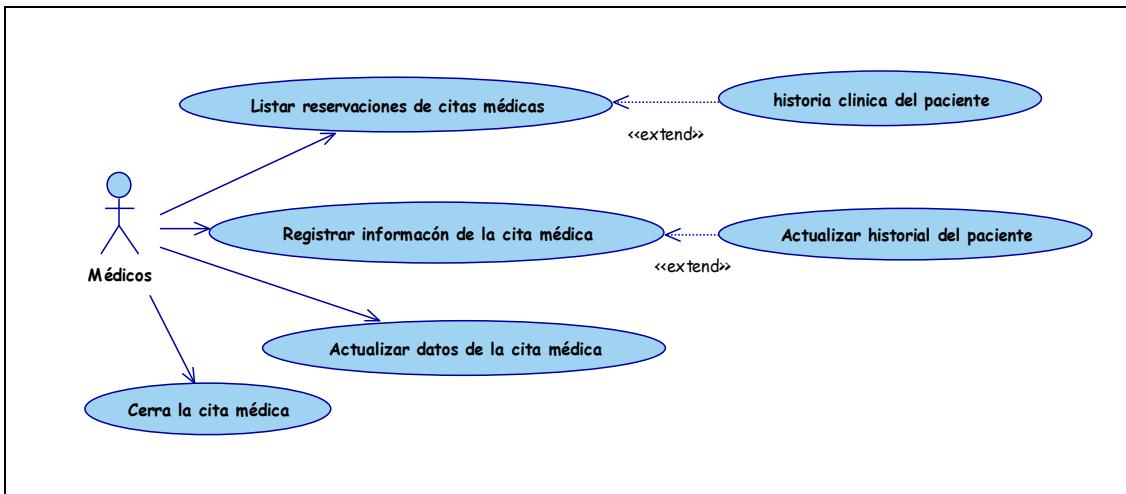
**Figura 3.9.** Caso de uso Listar reservaciones que deben ser atendidas por el médico.

Caso de uso:	Listar las reservaciones que deben ser atendidas por el profesional médico
Descripción:	Se despliega la lista de reservaciones que deben ser atendidas por el profesional médico.
Actores:	Profesional médico
Pre-Condiciones:	El usuario debe estar registrado en el sistema CEMOCITE. El usuario debe tener perfil de profesional médico.
Acciones básicas:	1. Ingresar a la aplicación desde el dispositivo. 2. Seleccionar el día de atención. 3. Listar reservaciones que deben ser atendidas por el profesional médico. 4. Ver historial clínico del paciente
Post-Condición:	Visualización de las reservaciones.

Fuente: Propia

**Tabla 3.21.** Caso de Uso Módulo usuarios: Listar reservaciones que deben ser atendidas por el médico.

- Caso de uso cita atendida por el profesional médico



Fuente: Propia

**Figura 3.10.** Caso de uso cita atendida por el profesional médico.

Caso de uso:	Cita atendida por el profesional médico
Descripción:	Se despliega la información del paciente en su historia clínica
Actores:	Profesional médico
Pre-Condiciones:	El paciente debe tener una cita confirmada a la fecha y hora elegidas.
Acciones básicas:	<ol style="list-style-type: none"> <li>1. Registrar la información de la cita</li> <li>2. Incluir observaciones</li> <li>3. Actualizar datos del historial clínico del paciente</li> <li>4. Cerrar cita.</li> </ol>
Post-Condición:	Actualización del historial clínico.

Fuente: Propia

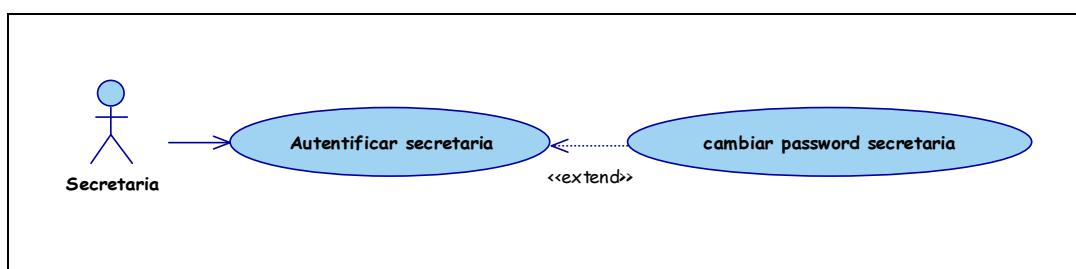
**Tabla 3.22.** Caso de Uso Módulo usuarios: cita atendida por el profesional médico.

### CASOS DE USO DEL ACTOR: SECRETARIA

Los casos de uso se usan para especificar el comportamiento del sistema

CEMOCITE con la secretaria son:

- Caso de uso Autenticar secretaria



Fuente: Propia

**Figura 3.11.** Caso de uso Autenticar secretaria

Caso de uso:	Autentificar secretaria
Descripción:	Permite ingresar a CEMOCITE mediante un login y un password desde el dispositivo.
Actores:	Secretaria
Pre-Condiciones:	El usuario debe estar registrado en el sistema CEMOCITE.
Acciones básicas:	<ol style="list-style-type: none"> <li>Conexión con el sistema CEMOCITE.</li> <li>Ingresar login y password.</li> </ol>
Acciones alternativas:	Si el login o password no son válidos, muestra un mensaje de error.
Post-Condición:	El sistema permite el acceso a la aplicación.

Fuente: Propia

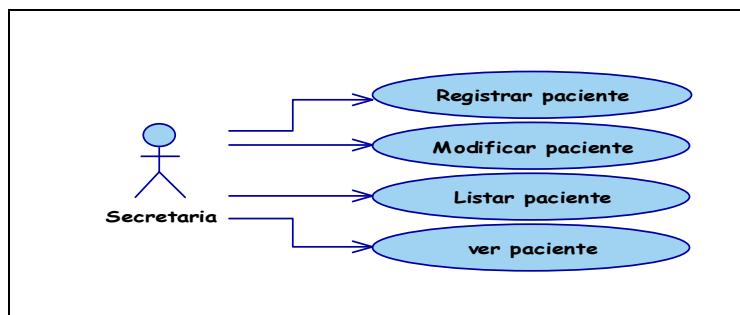
**Tabla 3.23.** Caso de Uso Módulo usuarios: Autentificar secretaria

Caso de uso:	Cambiar password secretaria
Descripción:	Permite cambiar el password de la secretaria
Actores:	Secretaria
Acciones básicas:	<ol style="list-style-type: none"> <li>Conectar al sistema CEMOCITE.</li> <li>Cambiar password</li> </ol>
Acciones alternativas:	Si el login o password no son válidos, el sistema muestra un mensaje de error.
Post-Condición:	El sistema procesa el cambio de password y Permite acceso al sistema

Fuente: Propia

**Tabla 3.24.** Caso de Uso Módulo usuarios: Cambiar password médico.

- Caso de uso Ingresar paciente**



Fuente: Propia

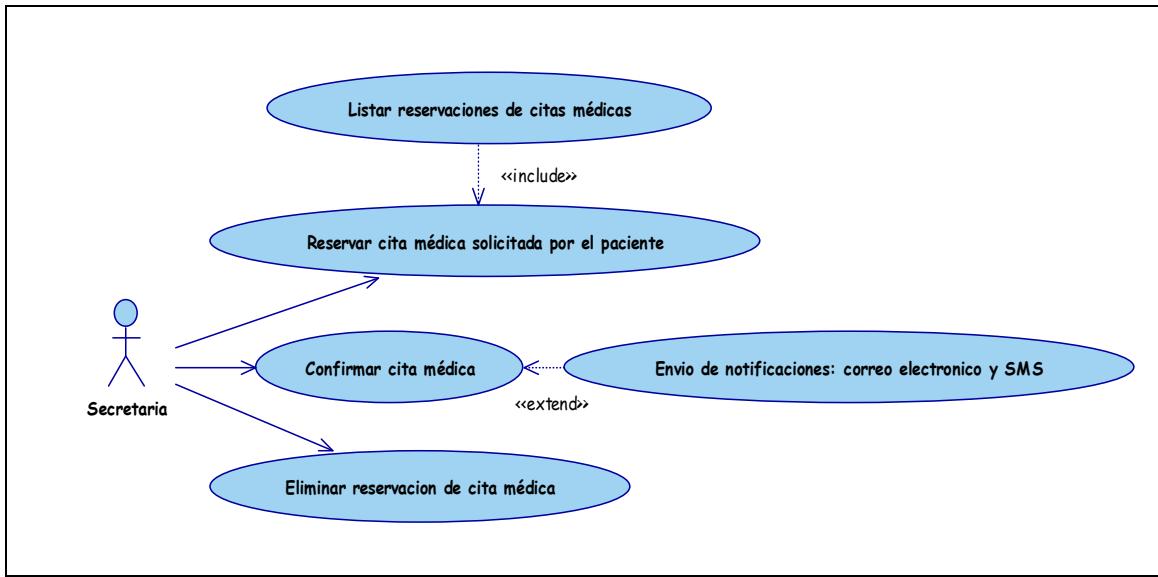
**Figura 3.12.** Caso de uso Ingresar paciente

Caso de uso:	Registrar, Modificar, listar y ver pacientes
Descripción:	Permite el registro y la modificación de los usuarios por medio del sitio Web.
Actores:	Secretaria.
Pre-Condiciones:	La secretaria solicita los datos a los usuarios para registrar en la aplicación web.
Acciones básicas:	<ol style="list-style-type: none"> <li>Conexión con el sitio Web CEMOCITE.</li> <li>Ingresar los datos necesarios del usuario.</li> <li>Guarda la información ingresada.</li> <li>Si se necesita hacer algún cambio modifica y guarda la información</li> </ol>
Acciones alternativas:	Si uno de los datos no son válidos, el sistema muestra un mensaje de error.
Post-Condición:	CEMOCITE devuelve la confirmación del registro creado permitiendo el acceso con su login y password registrados, la secretaria puede listar y ver los datos individuales de cada usuario

Fuente: Propia

**Tabla 3.25.** Caso de Uso Módulo usuarios : Registrar, modificar, listar y ver pacientes

- Caso de uso reservar cita médica solicitada por el paciente



Fuente: Propia

**Figura 3.13.** Caso de uso reservar cita médica solicitada por el paciente

Caso de uso:	Reservar cita médica solicitada por el paciente
Descripción:	Ingresar los datos de la reservación del paciente.
Actores:	Secretaria.
Pre-Condiciones:	La secretaria debe estar registrada en el sistema CEMOCITE. Debe tener perfil de secretaria, para poder reservar la cita para el paciente.
Acciones básicas:	<ol style="list-style-type: none"> <li>1. Ingresar a la aplicación desde el dispositivo.</li> <li>2. Solicitar turno.</li> <li>3. Seleccionar especialidad, médico y el día de atención.</li> <li>4. Notificar al paciente de la reservación.</li> </ol>
Post-Condición:	Visualización de las reservaciones.

Fuente: Propia

**Tabla 3.26.** Caso de Uso Módulo usuarios: Reservar cita médica solicitada por el paciente

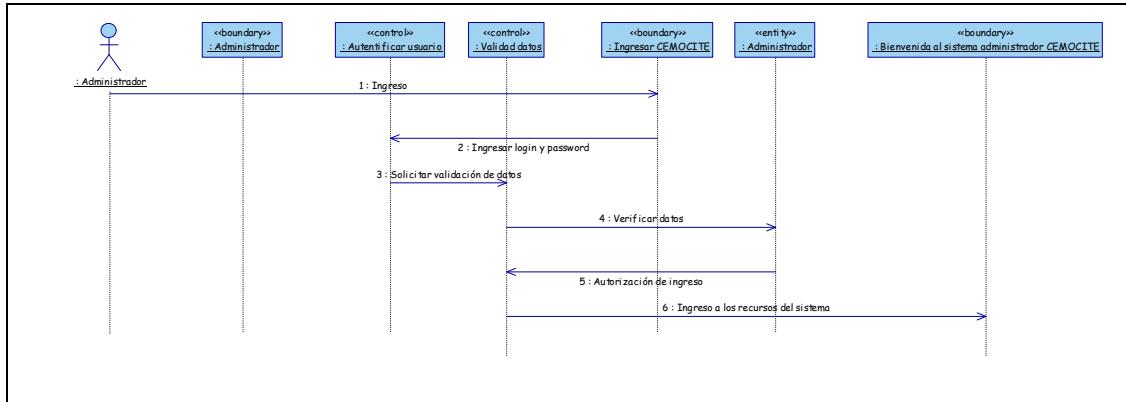
## 3.5 ANALISIS Y DISEÑO

### 3.5.1 Modelo de análisis

Establece la realización de los casos de uso en clases y pasa desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación), de acuerdo al avance del proyecto. Los diagramas de secuencias muestran la interacción de un conjunto de objetos en una aplicación a través del tiempo. Esta descripción es importante porque puede dar detalle a los casos de uso, aclarando el nivel de mensajes de los objetos existentes, como también muestra el uso de los mensajes de las clases diseñadas en el contexto de una operación.

## DIAGRAMAS DE SECUENCIAS DE LA REALIZACIÓN DE CASOS DE USO DEL ACTOR: ADMINISTRADOR

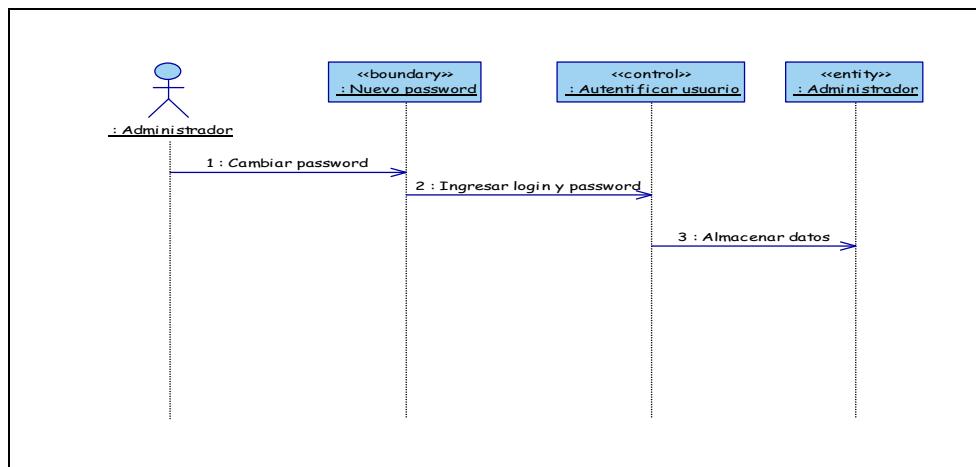
- **Diagrama de Secuencia del Cu: Autentificar usuario Administrador**



Fuente: Propia

**Figura 3.14.** Diagrama de secuencia del CU: Autentificar usuario administrador

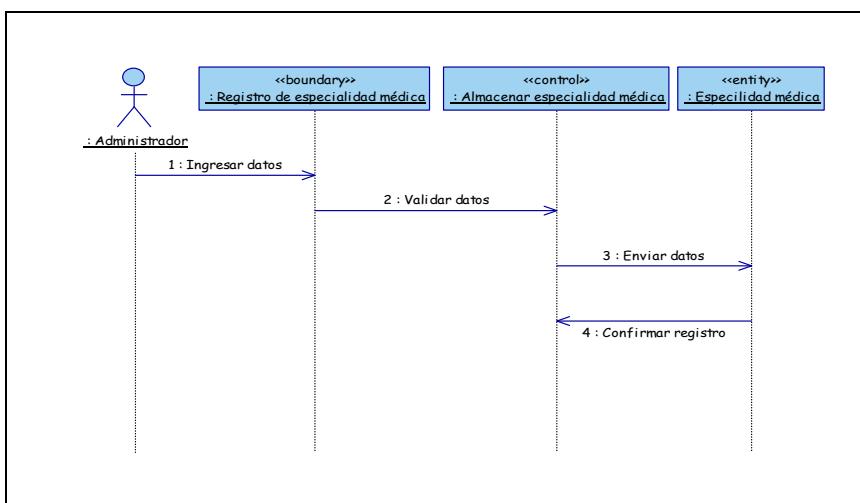
- **Diagrama de Secuencia del Cu: Cambiar password administrador**



Fuente: Propia

**Figura 3.15.** Diagrama de secuencia del CU: cambiar password usuario administrador

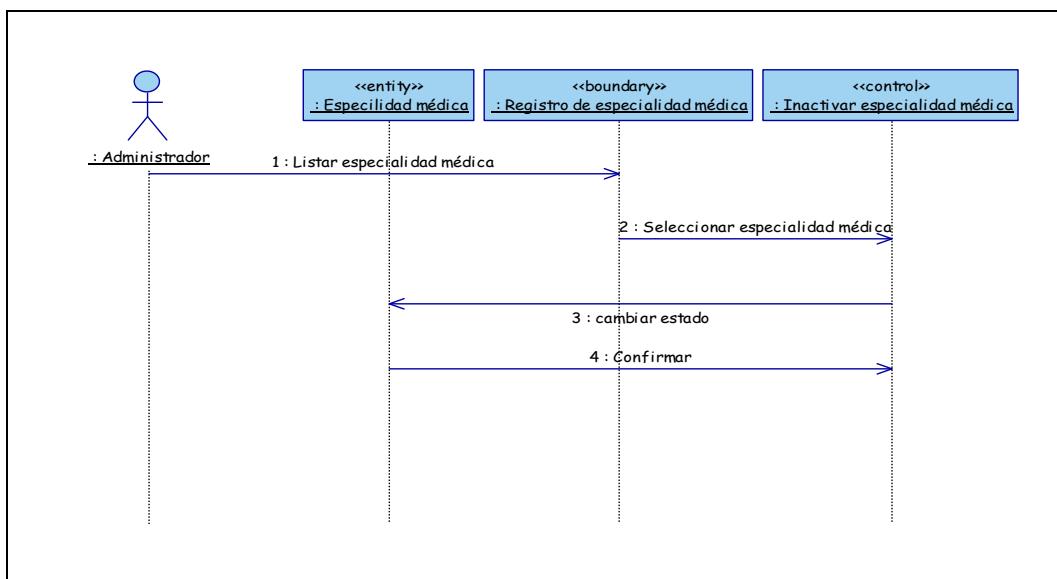
- **Diagrama de Secuencia del Cu: Registrar especialidad médica**



Fuente: Propia

**Figura 3.16.** Diagrama de secuencia del CU: Registrar especialidad médica

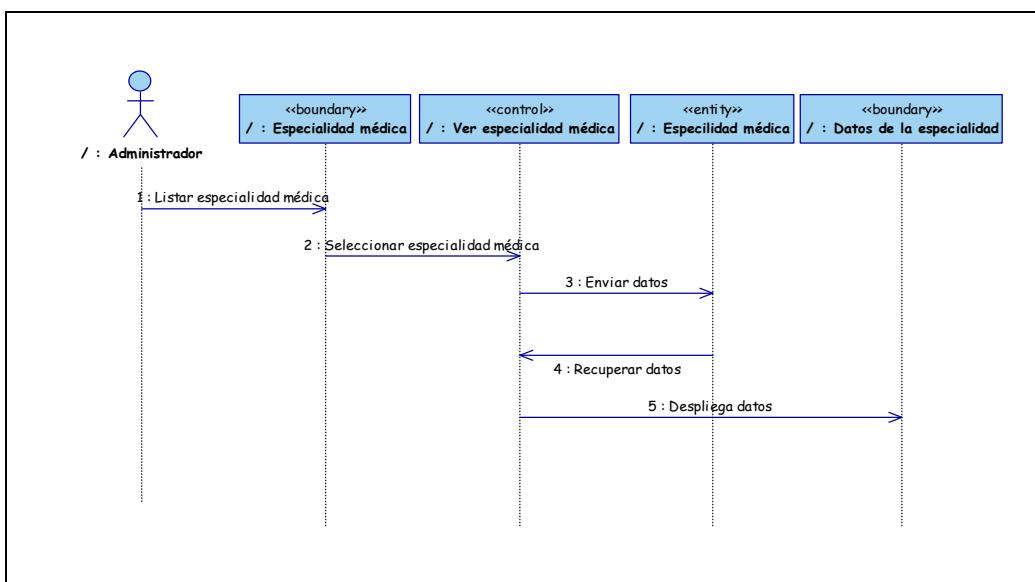
- **Diagrama de Secuencia del Cu: cambiar estado de especialidad médica**



Fuente: Propia

**Figura 3.17.** Diagrama de secuencia del CU: Cambiar estado de la especialidad médica

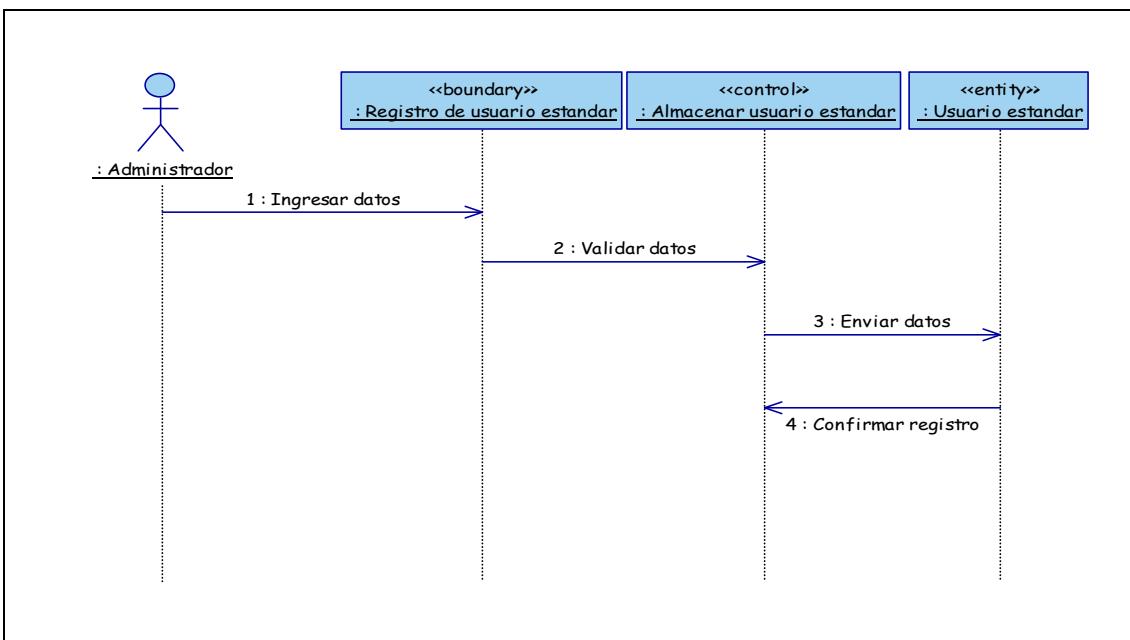
- **Diagrama de Colaboración del Cu: Listar especialidad médica**



Fuente: Propia

**Figura 3.18.** Diagrama de secuencia del CU: Listar especialidad médica

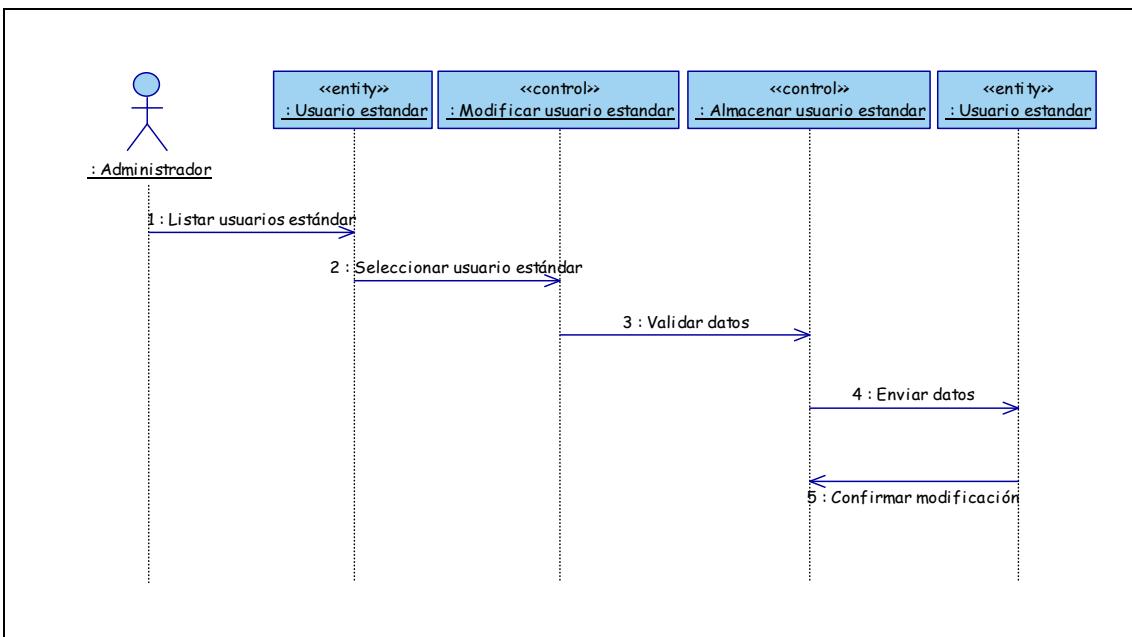
- **Diagrama de Secuencia del Cu: Registrar usuario estándar**



Fuente: Propia

**Figura 3.19.** Diagrama de secuencia del CU: Registrar Usuario estándar

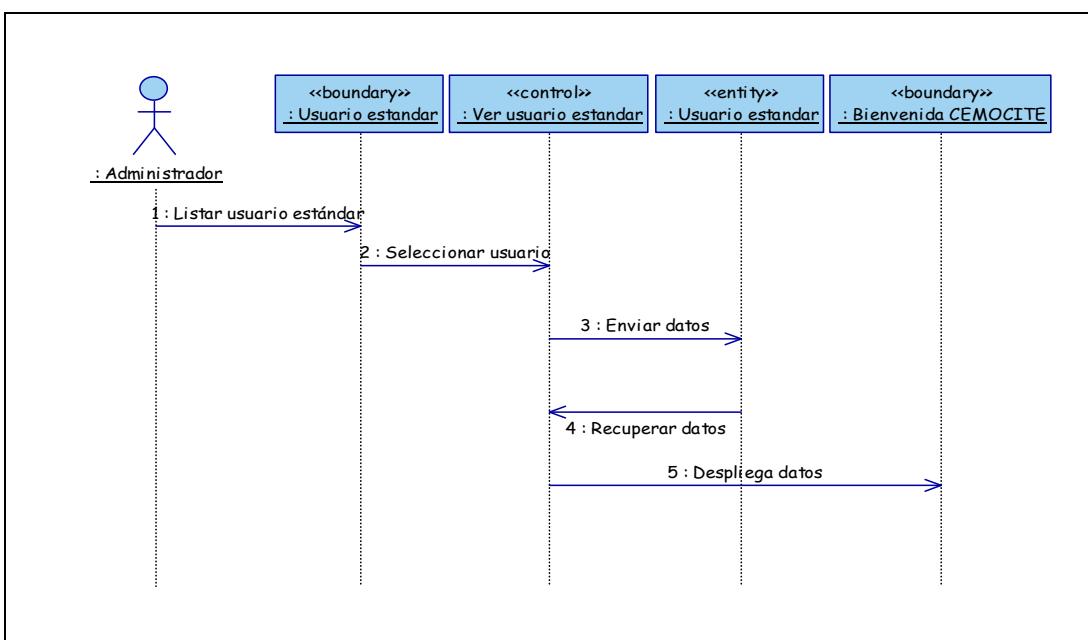
- **Diagrama de Colaboración del Cu: Modificar usuario estándar**



Fuente: Propia

**Figura 3.20.** Diagrama de secuencia del CU: Modificar Usuario estándar

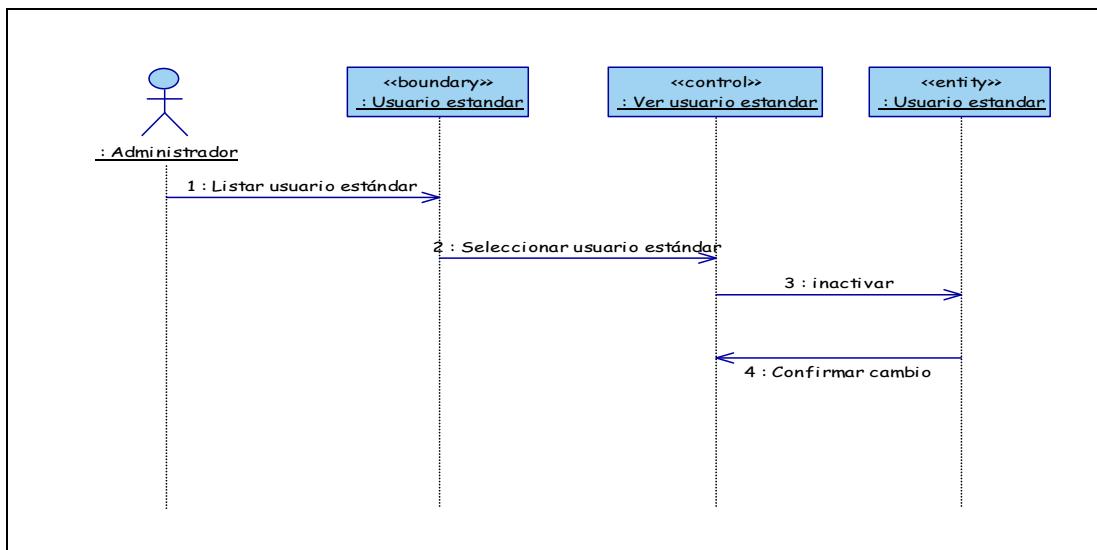
- **Diagrama de Secuencia del Cu: Listar usuario estándar**



Fuente: Propia

**Figura 3.21.** Diagrama de secuencia del CU: Listar Usuario estándar

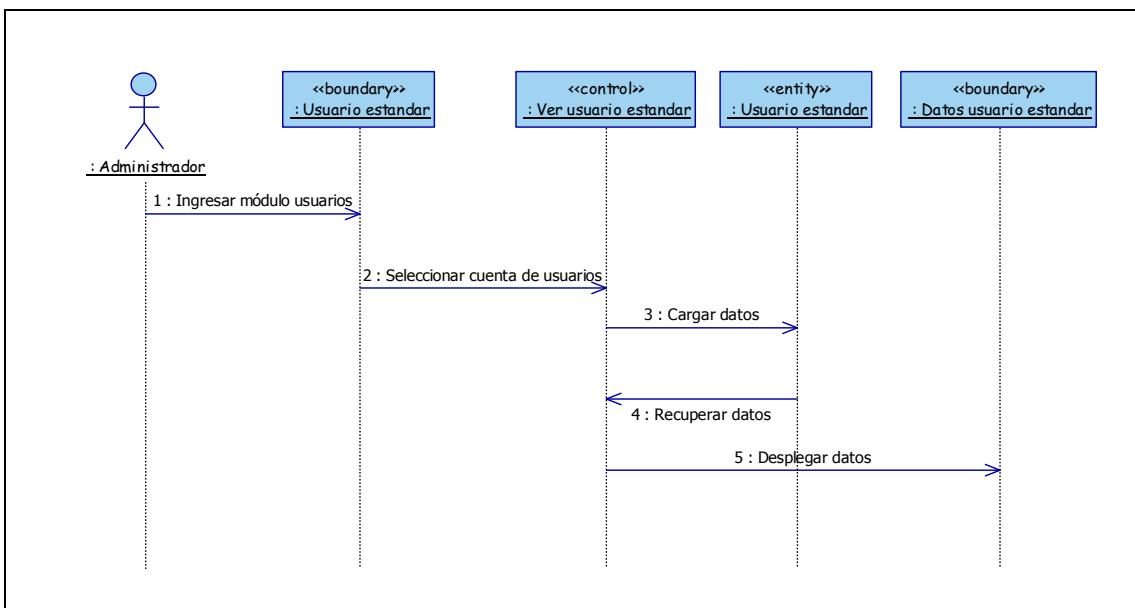
- **Diagrama de Secuencia del Cu: Inactivar usuario estándar**



Fuente: Propia

**Figura 3.22.** Diagrama de secuencia del CU: Eliminar Usuario estándar

- **Diagrama de Colaboración del Cu: Ver usuario estándar**

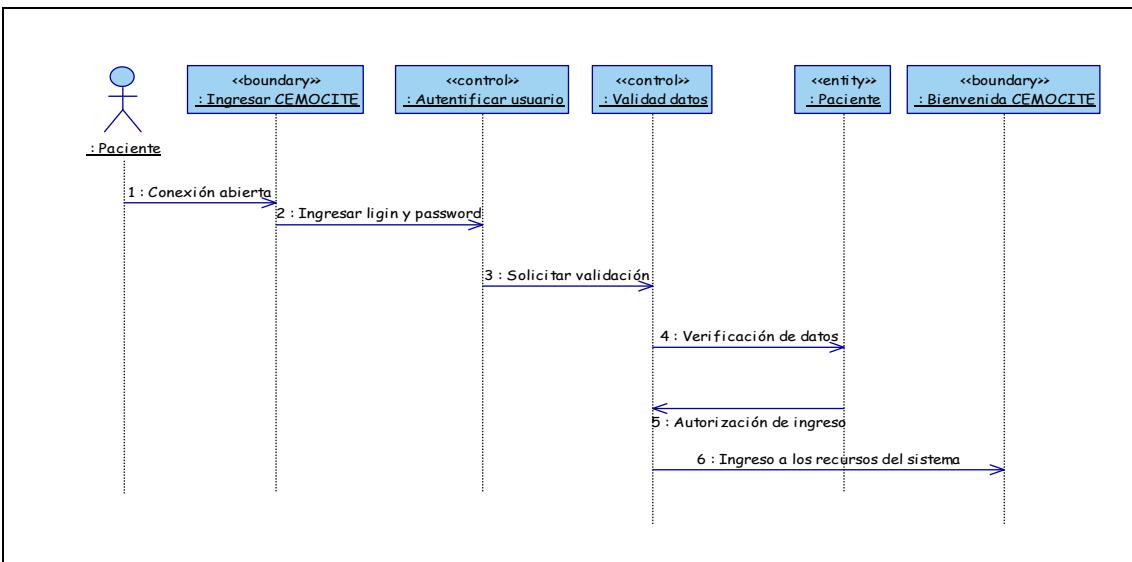


Fuente: Propia

**Figura 3.23.** Diagrama de secuencia del CU: Ver Usuario estándar

## DIAGRAMAS DE SECUENCIA DE LA REALIZACIÓN DE CASOS DE USO DEL ACTOR: PACIENTE

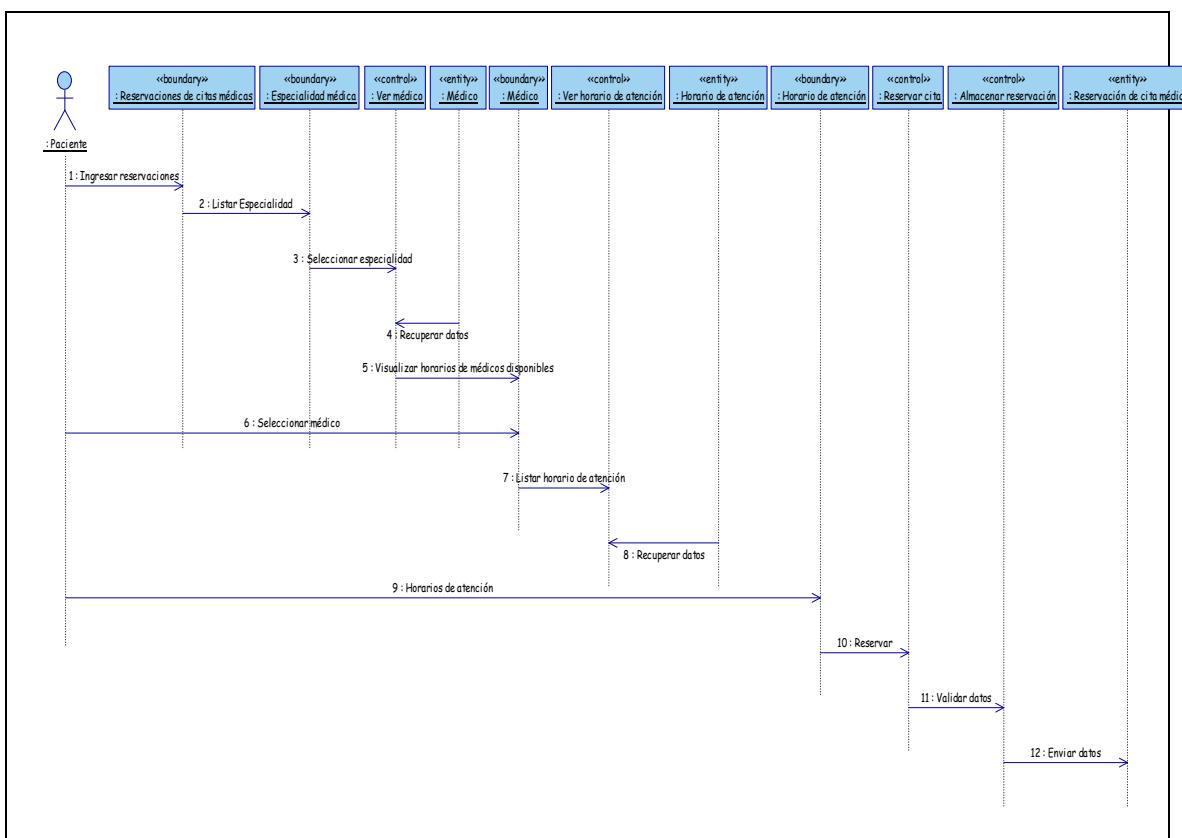
- *Diagrama de Secuencia del Cu: Autentificar paciente*



Fuente: Propia

**Figura 3.24.** Diagrama de secuencia del CU: Autentificar paciente

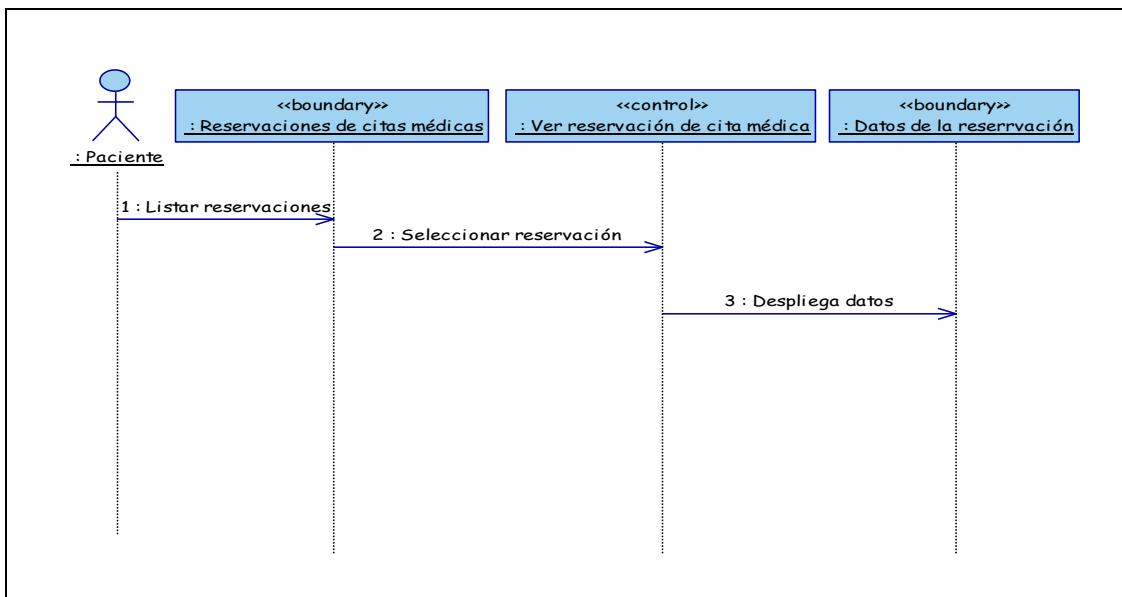
- *Diagrama de Colaboración del Cu: Reservar cita médica*



Fuente: Propia

**Figura 3.25.** Diagrama de secuencia del CU: Reservar cita médica

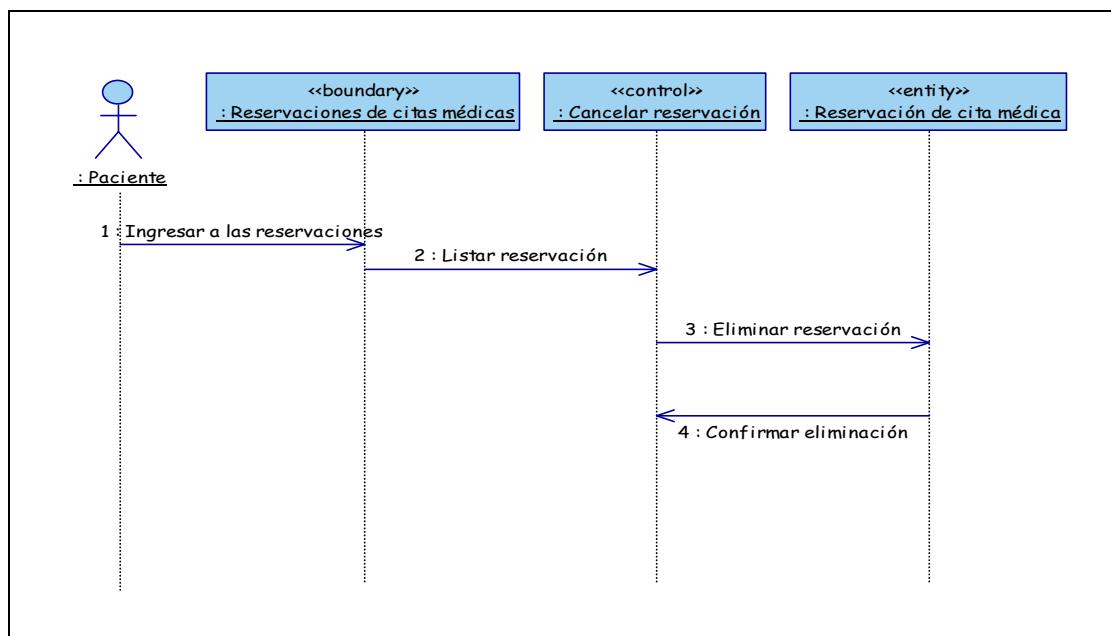
- **Diagrama de Colaboración del Cu: Ver Reservación de cita médica**



Fuente: Propia

**Figura 3.26.** Diagrama de secuencia del CU: Ver Reservación de cita médica

- **Diagrama de Colaboración del Cu: Cancelar Reservación de cita médica**

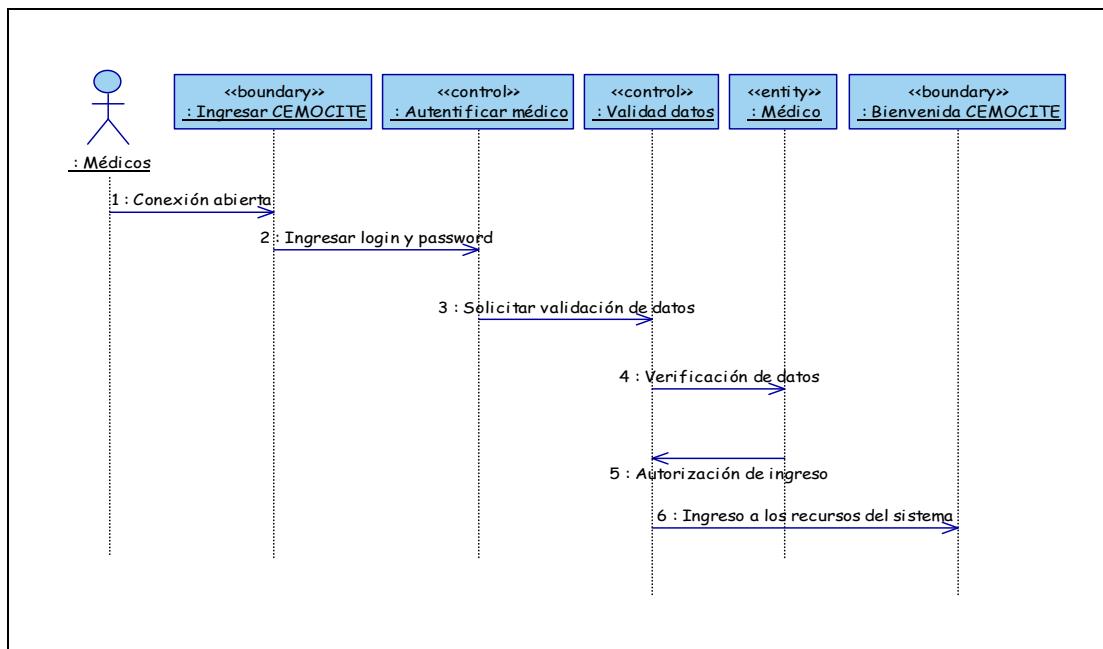


Fuente: Propia

**Figura 3.27.** Diagrama de secuencia del CU: Cancelar cita médica

## DIAGRAMAS DE SECUENCIA DE LA REALIZACIÓN DE CASOS DE USO DEL ACTOR: MÉDICO

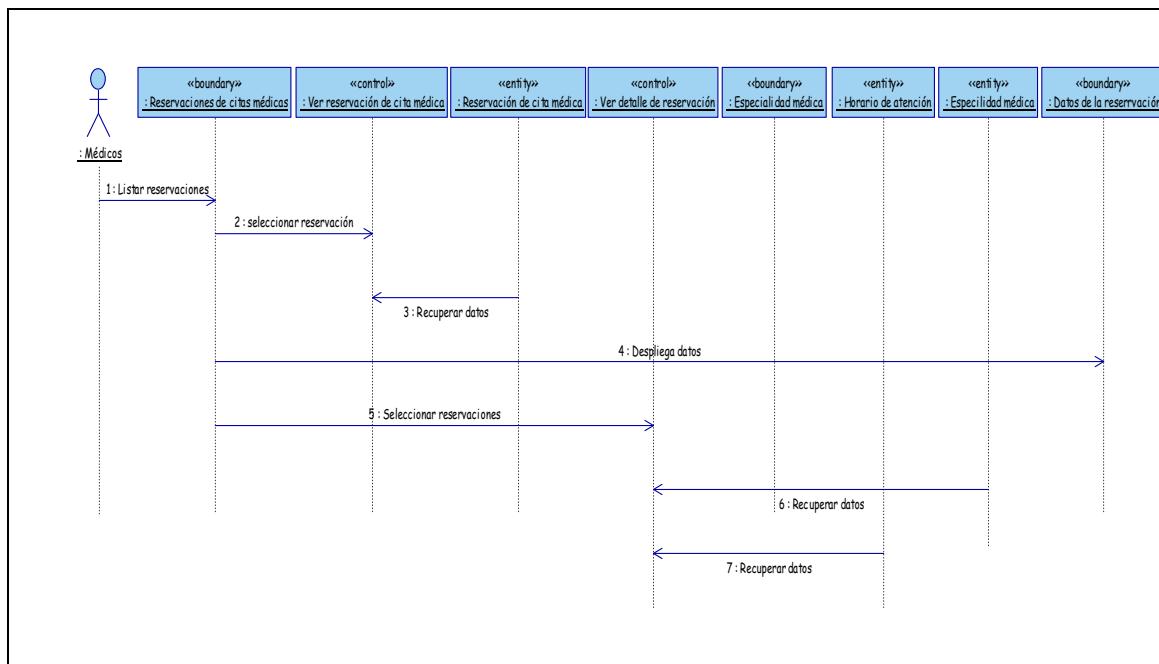
- **Diagrama de Secuencia del Cu: Autentificar médico**



Fuente: Propia

**Figura 3.28.** Diagrama de secuencia del CU: Autentificar médico

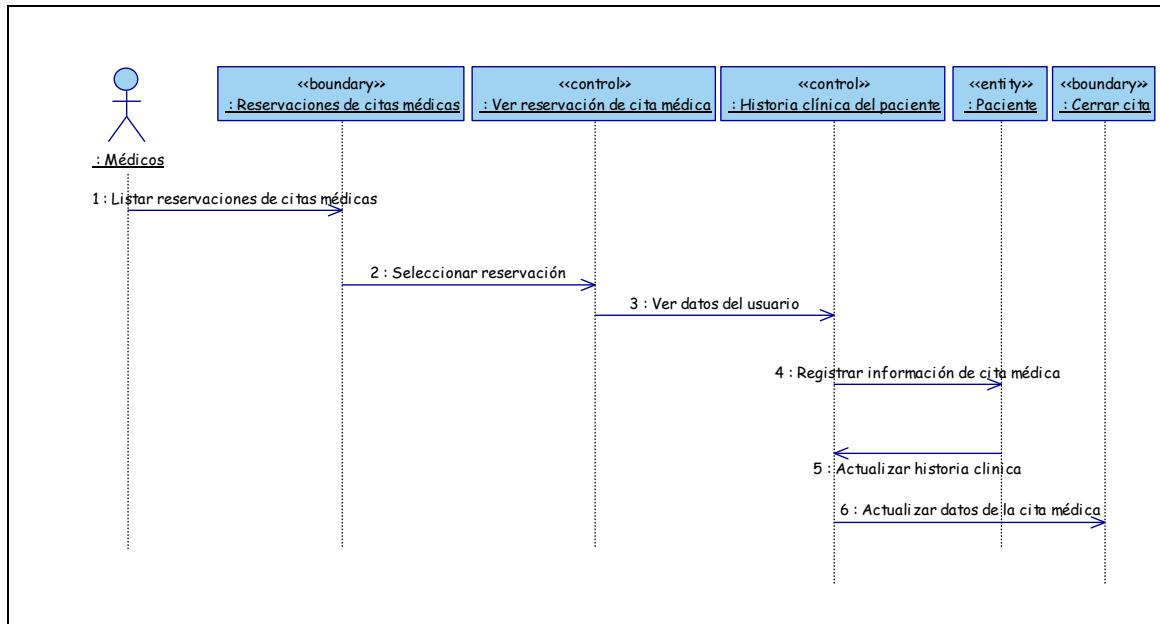
- **Diagrama de Secuencia del Cu: Listar reservaciones que deben ser atendidas por el profesional médico**



Fuente: Propia

**Figura 3.29.** Diagrama de secuencia del CU: Listar reservaciones que deben ser atendidas por el profesional médico

- **Diagrama de Secuencia del Cu: Cita atendida por el profesional médico**

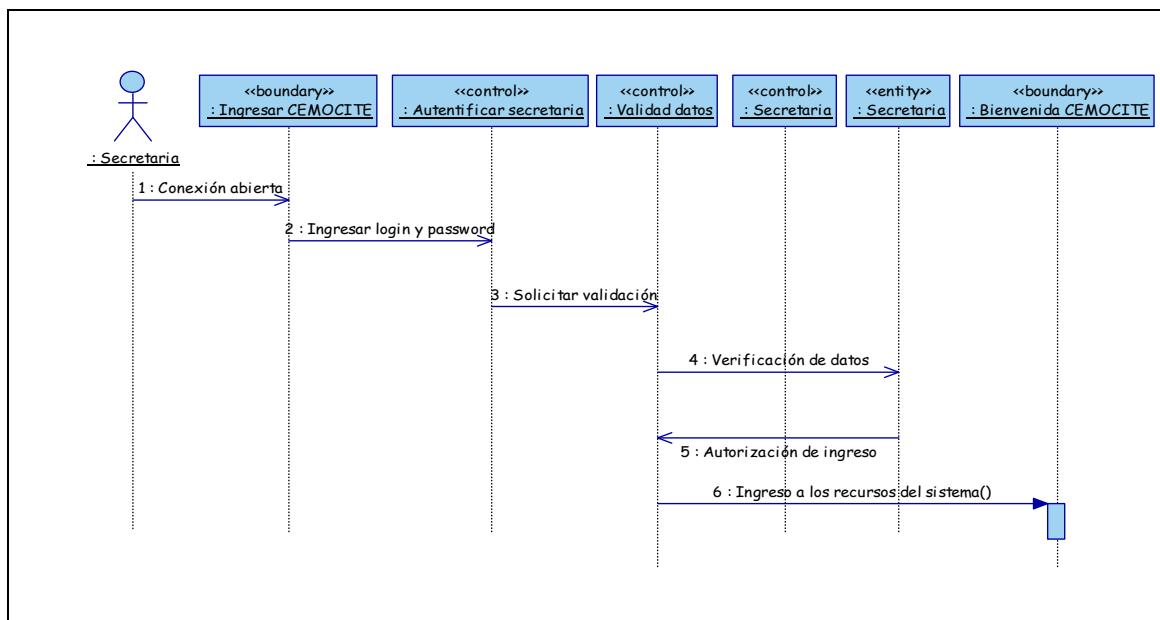


Fuente: Propia

**Figura 3.30.** Diagrama de secuencia del CU: Cita atendida por el profesional médico

## DIAGRAMAS DE SECUENCIA DE LA REALIZACIÓN DE CASOS DE USO DEL ACTOR:SECRETARIA

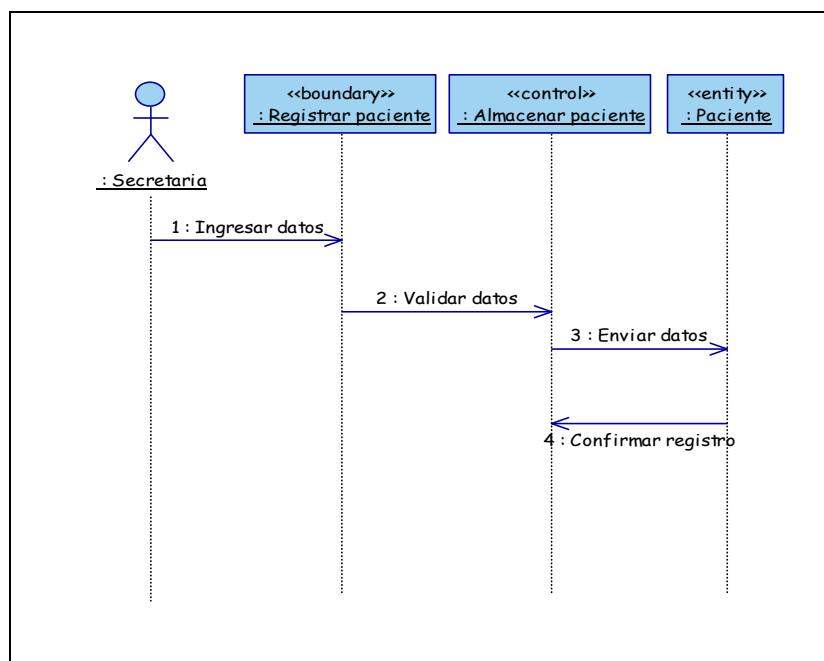
- **Diagrama de Secuencia del Cu: Autentificar secretaria**



Fuente: Propia

**Figura 3.31.** Diagrama de secuencia del CU: Autentificar secretaria

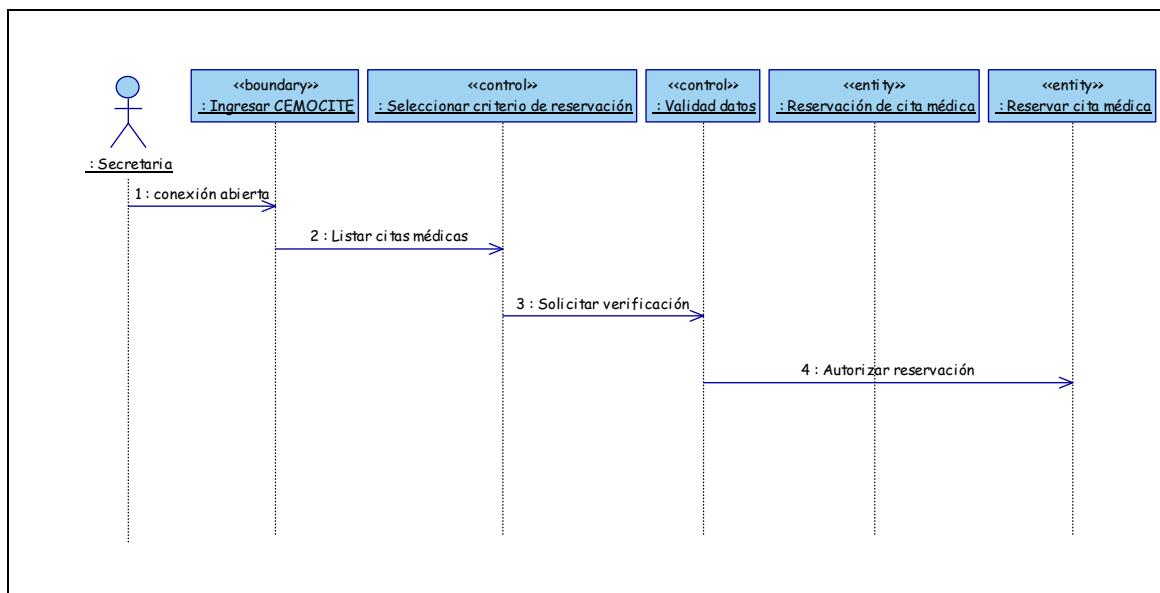
- **Diagrama de Secuencia del Cu: Ingresar paciente**



Fuente: Propia

**Figura 3.32.** Diagrama de secuencia del CU: Ingresar paciente

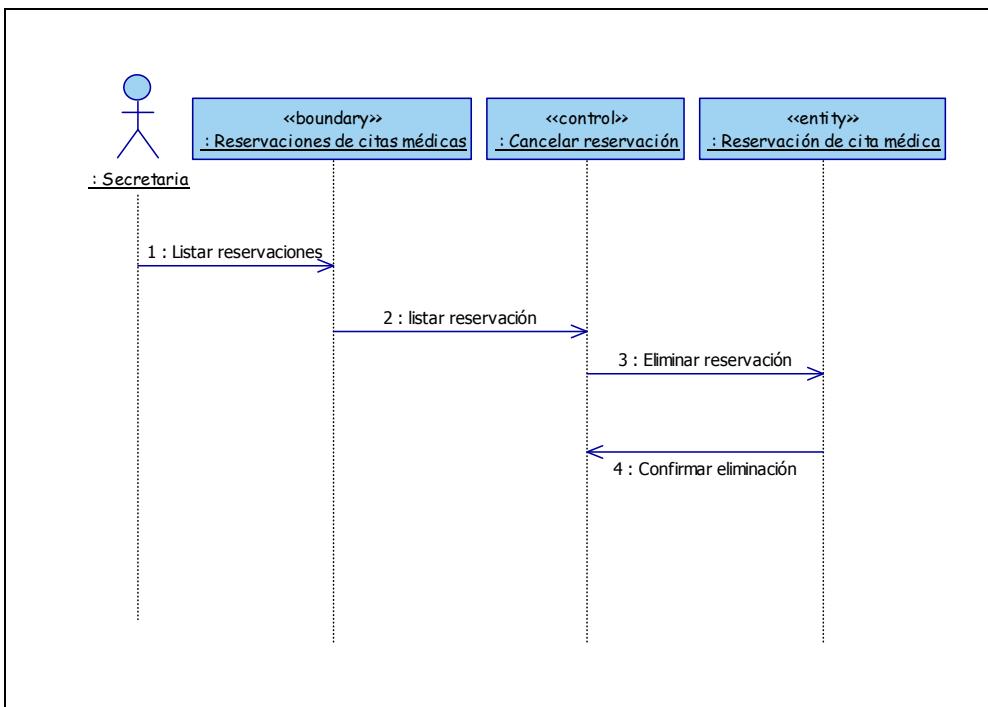
- **Diagrama de Secuencia del Cu: Reservar cita médica solicitada por el paciente**



Fuente: Propia

**Figura 3.33.** Diagrama de secuencia del CU: Reservar cita médica solicitada por el paciente

- **Diagrama de Secuencia del Cu: Cancelar reservación de cita médica**



Fuente: Propia

**Figura 3.34.** Diagrama de secuencia del CU: Cancelar cita médica

## PAQUETES DE ANALISIS

Los paquetes de análisis son particionamientos funcionales del sistema basados en el dominio del problema, permitiendo ser un medio de organización de clases de análisis, realizaciones de casos de uso, y otros paquetes del modelo de análisis.

La identificación de los paquetes de análisis se basan en los requisitos funcionales y en el dominio del problema, agrupando un cierto número de casos de uso en un paquete determinado, y la realización de la funcionalidad correspondiente dentro de dicho paquete.

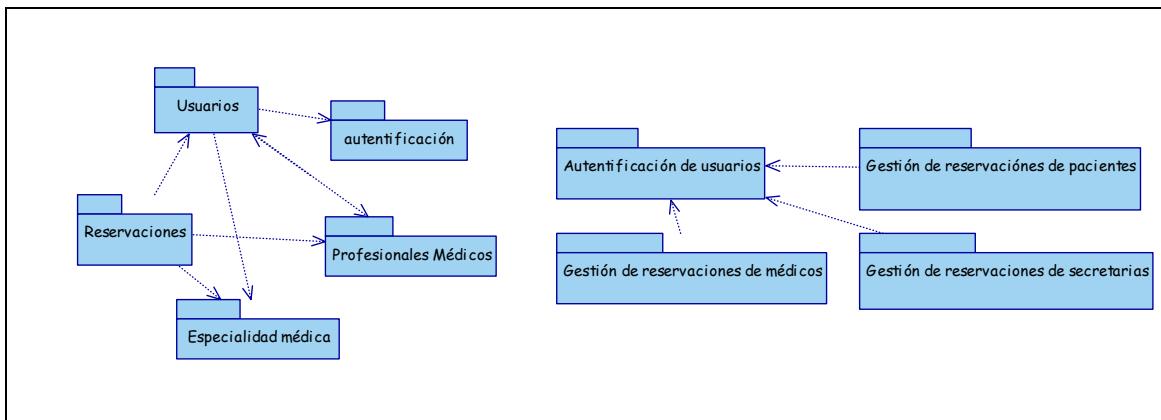
De acuerdo a estas consideraciones se han identificado los paquetes de análisis del sistema CEMOCITE, bajo este concepto, los paquetes de análisis que se han identificado para CEMOCITE se especifican en la siguiente tabla.

Módulo	Paquete de Análisis	Casos de uso
Administrador	Autentificación	Autenticar usuario administrador. Cambiar password del administrador
Administrador	Gestión de especialidades médicas	Registrar especialidad médica Ver especialidad médica Modificar especialidad médica Listar especialidad médica Inactivar especialidad médica
Administrador	Gestión de usuarios (Médico secretaria pacientes)	Autenticar usuario Registrar usuario Ver usuario Listar usuarios Modificar usuario Inactivar usuario
Administrador	Reservaciones de citas médicas	Listar reservaciones citas médicas Eliminar reservaciones de citas médicas Ver citas médicas
Paciente	Gestión de reservaciones del Usuario paciente	Autenticar usuario Reservación de citas médicas Seleccionar criterio de reservación Reservar cita médica Listar reservaciones de citas médicas. Ver reservación de cita médica Cancelar reservaciones de citas médicas.
Médico	Gestión de reservaciones profesional médico	Autenticar profesional médico Listar reservaciones médicas que deben ser atendidas por el profesional médico.
Secretaria	Gestión de reservaciones secretaria	Autenticar secretaria Registrar paciente Reservar cita médica solicitada por el paciente. Eliminar cita médica

Fuente: Propia

**Tabla 3.27. Identificación de paquetes de análisis**

La relación de dependencia de los paquetes de análisis se visualiza en la siguiente figura:



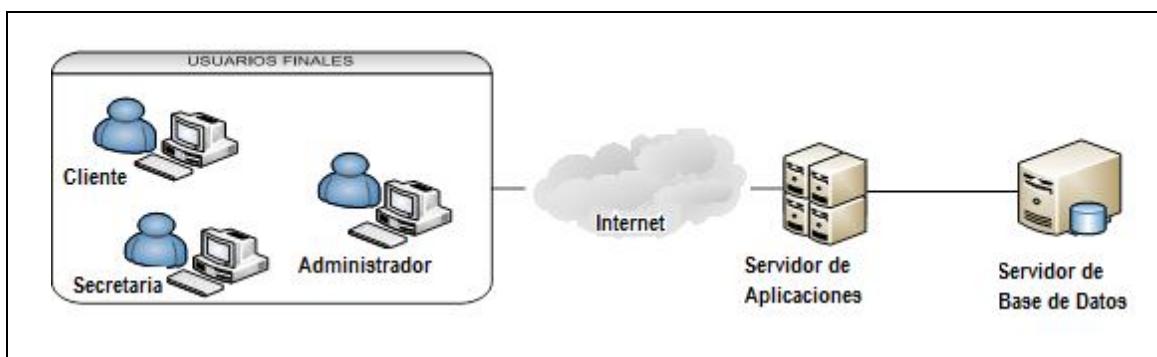
Fuente: Propia

**Figura 3.35.** Dependencia entre paquetes de análisis

En la etapa del diseño se modela el sistema y su arquitectura de tal manera que soporte todos los requisitos funcionales y no funcionales. El diseño es el centro de atención al final de la fase de elaboración y comienzo de las iteraciones de construcción. Se toma como entrada el modelo de análisis

### **3.5.2 Arquitectura del sistema**

La arquitectura del sistema CEMOCITE abarca decisiones importantes sobre la organización del sistema, la selección de los elementos estructurales que compondrán el sistema y sus interfaces, así como también se tomará en cuenta el uso, funcionalidad, rendimiento, capacidad de adaptación, reutilización, capacidad para ser comprendida, restricciones.



Fuente: Propia

**Figura 3.36.** Arquitectura Planteada

En la figura 3.36 se detalla la organización del sistema y como cada uno de los elementos se interrelacionan entre sí; por un lado tenemos a los usuarios finales comprendidos por los roles que intervienen en el sistema a través de un navegador de internet, que al digitar la url del sistema permite la respuesta por el servidor de aplicaciones con los datos seleccionados desde el servidor de base de datos de acuerdo al rol que desempeña cada usuario final.

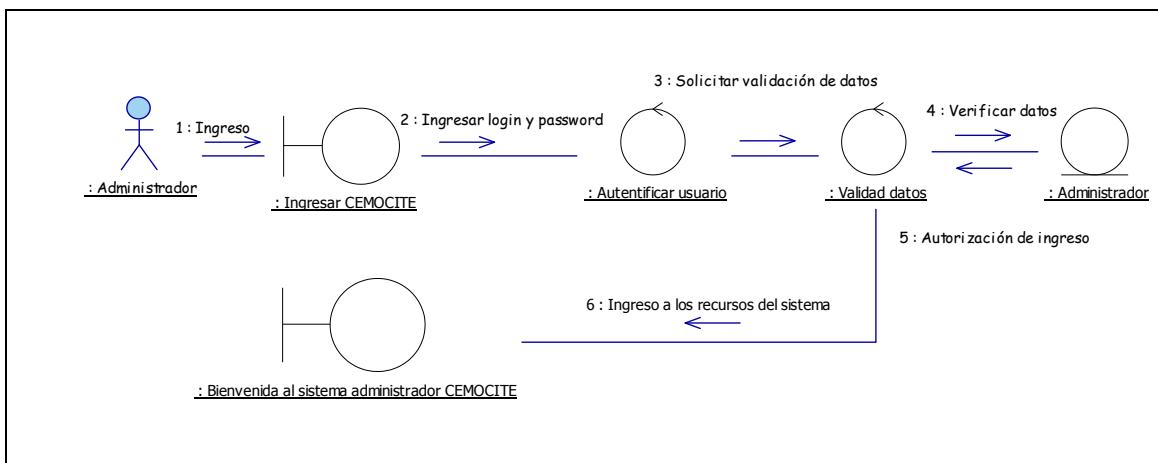
### **3.5.3 Diseño de casos de uso**

Los casos de uso pueden ser representados en la etapa de diseño mediante la representación de diagramas de colaboración, permitiendo una entrada a la implementación como un modelo estable para un mejor progreso del proyecto.

El sistema CEMOCITE es representado en esta etapa mediante los diagramas de Colaboración.

## DIAGRAMAS DE COLABORACIÓN DE LA REALIZACIÓN DE CASOS DE USO DEL ACTOR: ADMINISTRADOR

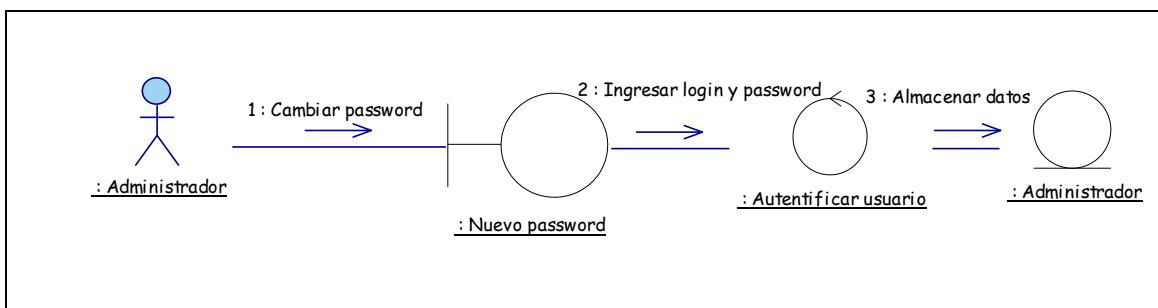
- **Diagrama de Colaboración del Cu: Autentificar usuario Administrador**



Fuente: Propia

**Figura 3.37.** Diagrama de colaboración del CU: Autentificar usuario administrador

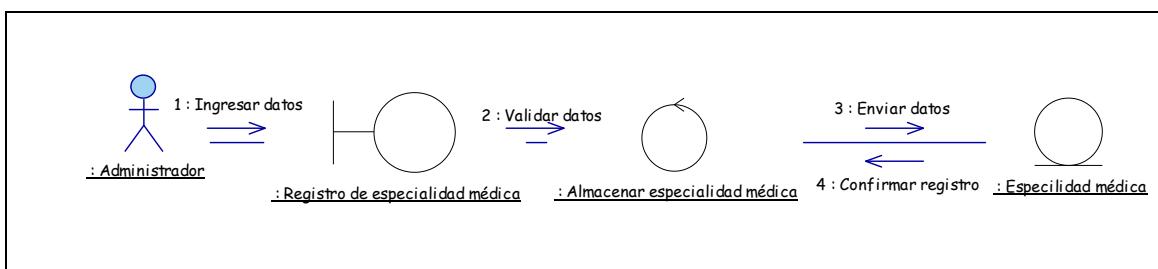
- **Diagrama de Colaboración del Cu: Cambiar password administrador**



Fuente: Propia

**Figura 3.38.** Diagrama de colaboración del CU: cambiar password usuario administrador

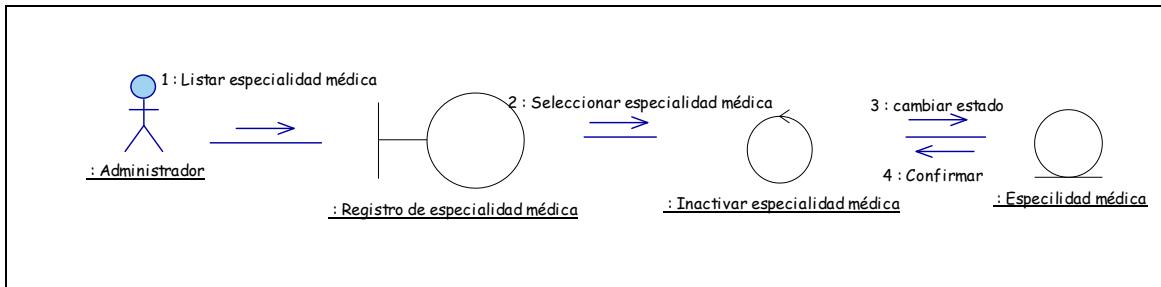
- **Diagrama de Colaboración del Cu: Registrar especialidad médica**



Fuente: Propia

**Figura 3.39.** Diagrama de colaboración del CU: Registrar especialidad médica

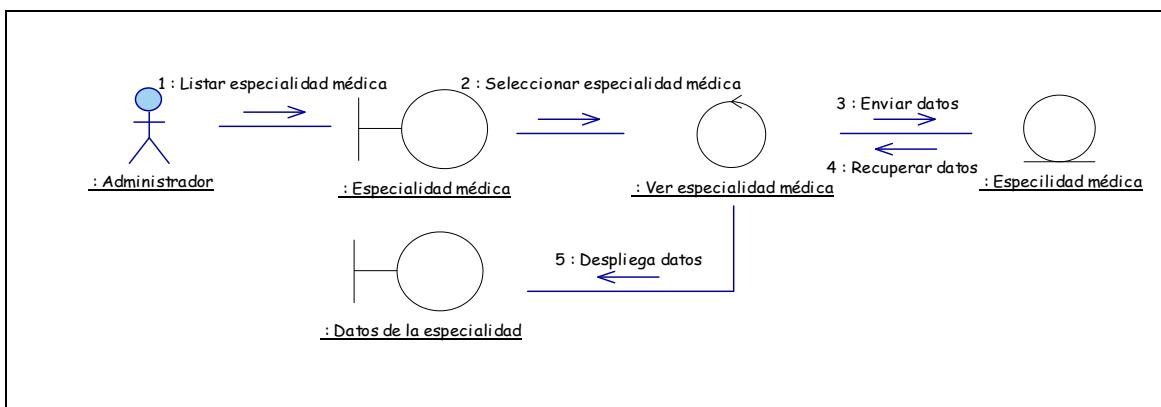
- **Diagrama de Colaboración del Cu: Cambiar estado especialidad médica**



Fuente: Propia

**Figura 3.40.** Diagrama de colaboración del CU: Eliminar especialidad médica

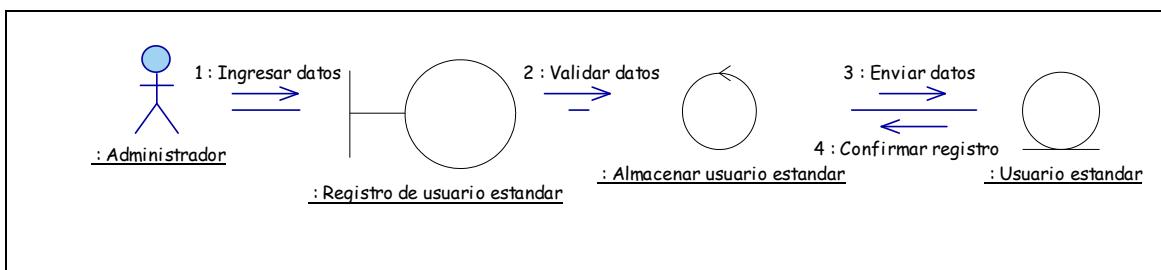
- **Diagrama de Colaboración del Cu: Listar especialidad médica**



Fuente: Propia

**Figura 3.41.** Diagrama de colaboración del CU: Listar especialidad médica

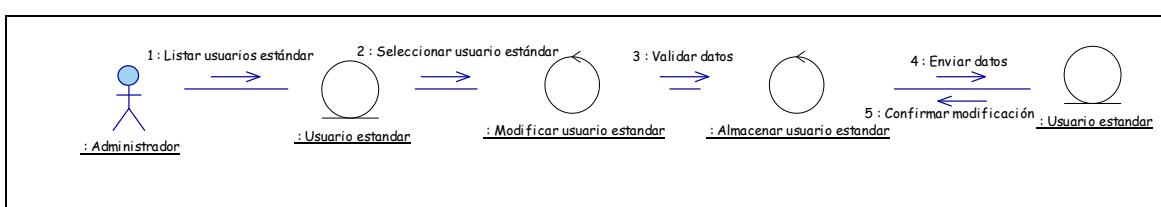
- **Diagrama de Colaboración del Cu: Registrar usuario estándar**



Fuente: Propia

**Figura 3.42.** Diagrama de colaboración del CU: Registrar Usuario estándar

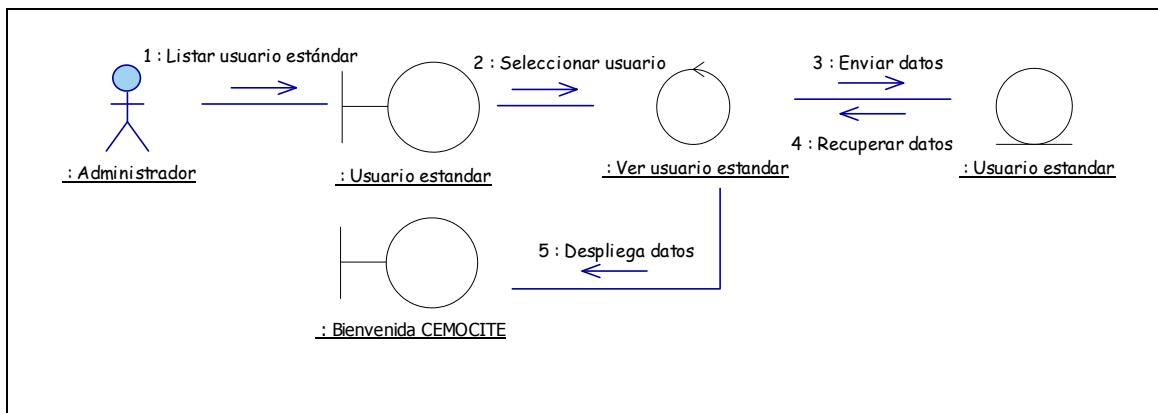
- **Diagrama de Colaboración del Cu: Modificar usuario estándar**



Fuente: Propia

**Figura 3.43.** Diagrama de colaboración del CU: Modificar Usuario estándar

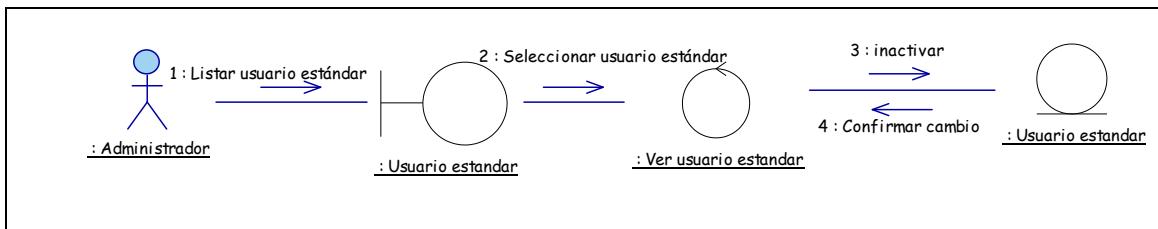
- **Diagrama de Colaboración del Cu: Listar usuario estándar**



Fuente: Propia

**Figura 3.44.** Diagrama de colaboración del CU: Listar Usuario estándar

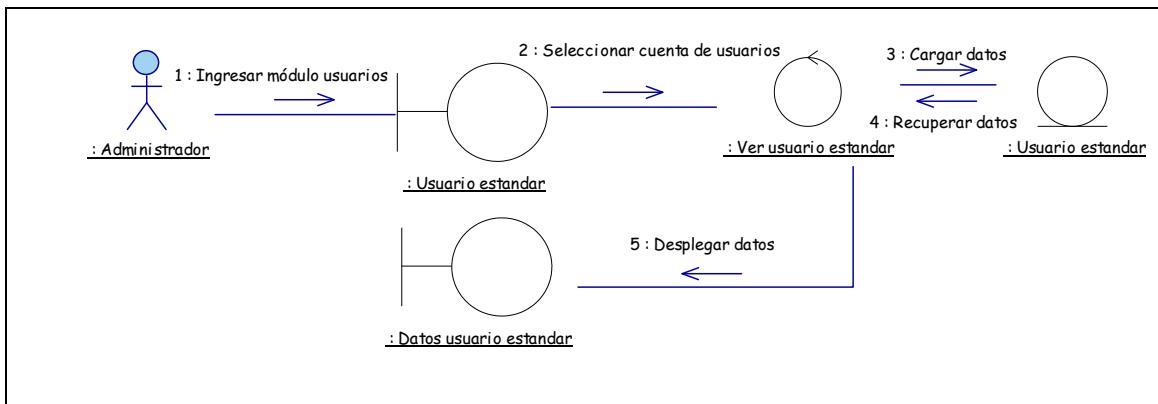
- **Diagrama de Colaboración del Cu: Cambiar de estado usuario estándar**



Fuente: Propia

**Figura 3.45.** Diagrama de colaboración del CU: Eliminar Usuario estándar

- **Diagrama de Colaboración del Cu: Ver usuario estándar**

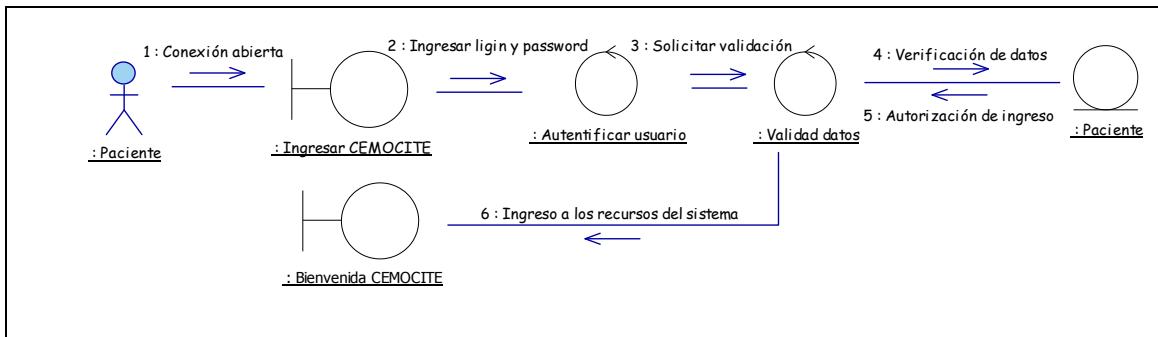


Fuente: Propia

**Figura 3.46.** Diagrama de colaboración del CU: Ver Usuario estándar

## DIAGRAMAS DE COLABORACIÓN DE LA REALIZACIÓN DE CASOS DE USO DEL ACTOR: PACIENTE

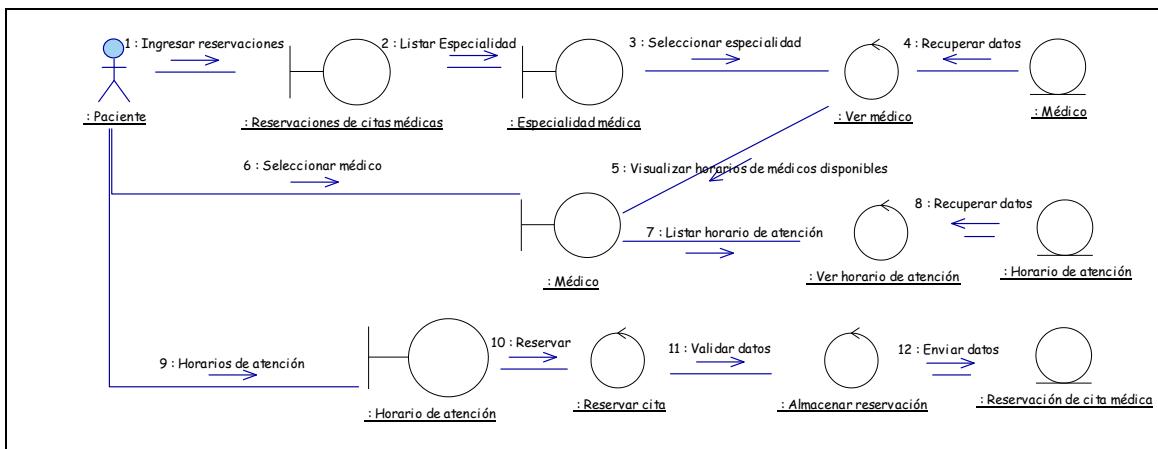
- *Diagrama de Colaboración del Cu: Autentificar paciente*



Fuente: Propia

**Figura 3.47.** Diagrama de colaboración del CU: Autentificar paciente

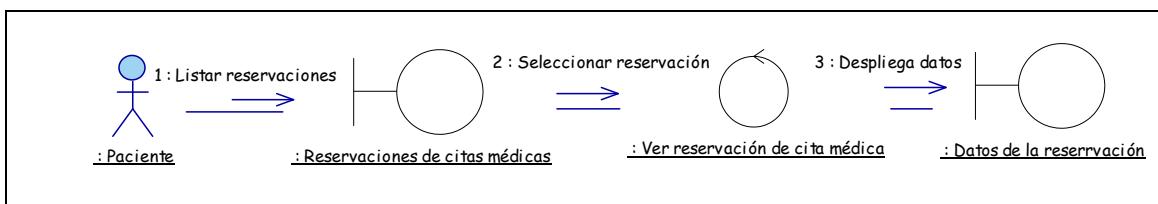
- *Diagrama de Colaboración del Cu: Reservar cita médica*



Fuente: Propia

**Figura 3.48.** Diagrama de colaboración del CU: Reservar cita médica

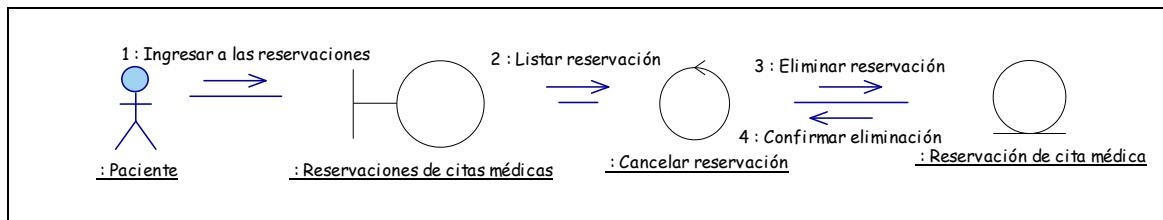
- *Diagrama de Colaboración del Cu: Ver reservación de cita médica*



Fuente: Propia

**Figura 3.49.** Diagrama de colaboración del CU: Ver Reservación de cita médica

- **Diagrama de Colaboración del Cu: Cancelar reservación de cita médica**

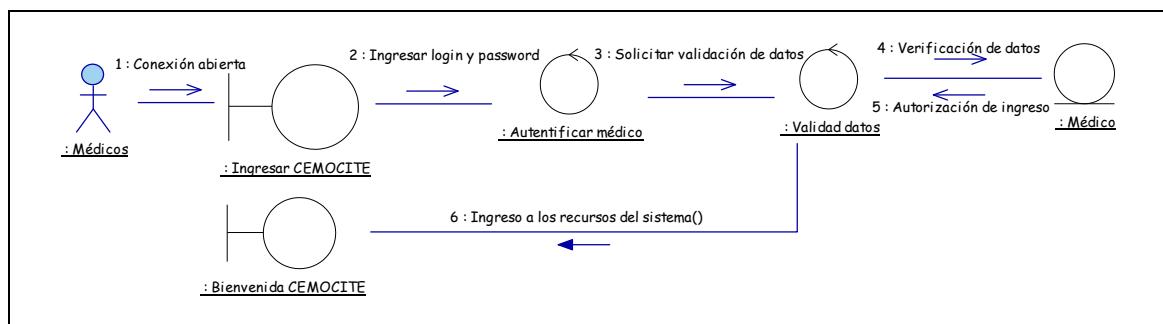


Fuente: Propia

**Figura 3.50.** Diagrama de colaboración del CU: Cancelar Reservación de cita médica

## DIAGRAMAS DE COLABORACIÓN DE LA REALIZACIÓN DE CASOS DE USO DEL ACTOR: MÉDICO

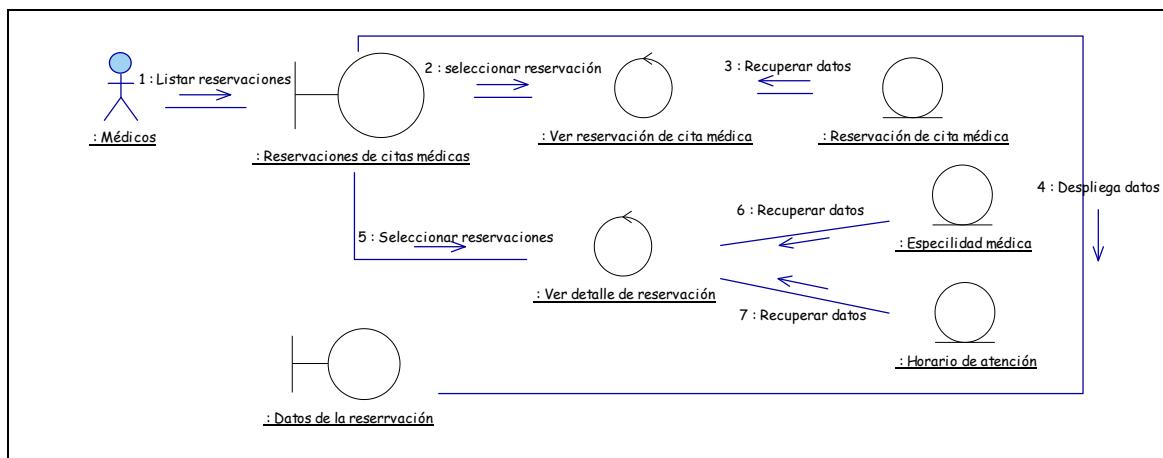
- **Diagrama de Colaboración del Cu: Autentificar médico**



Fuente: Propia

**Figura 3.51.** Diagrama de colaboración del CU: Autentificar médico

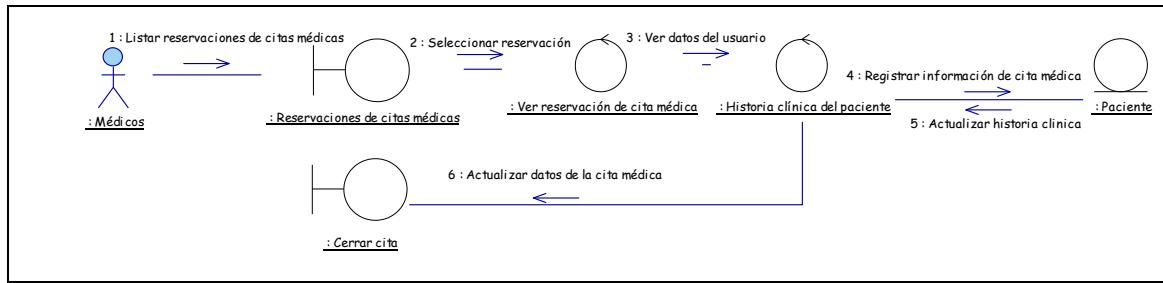
- **Diagrama de Colaboración del Cu: Listar reservaciones que deben ser atendidas por el profesional médico**



Fuente: Propia

**Figura 3.52.** Diagrama de colaboración del CU: Listar reservaciones que deben ser atendidas por el médico

- **Diagrama de Colaboración del Cu: Cita atendida por el profesional médico**

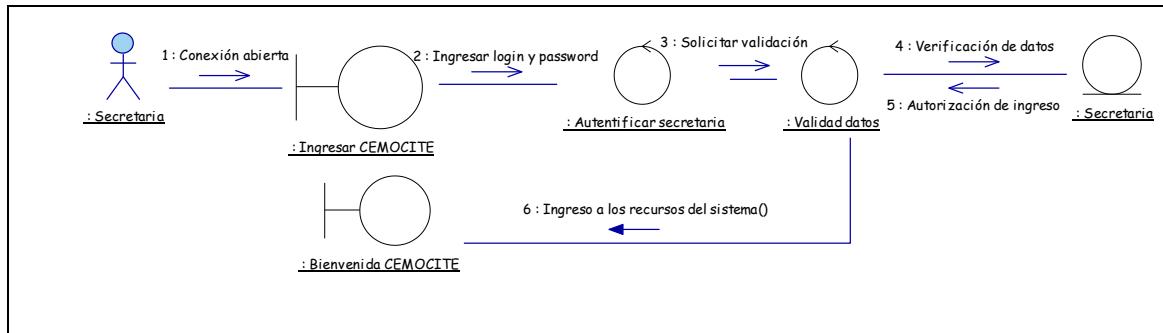


Fuente: Propia

**Figura 3.53.** Diagrama de colaboración del CU: Cita atendida por el profesional médico

## DIAGRAMAS DE COLABORACIÓN DE LA REALIZACIÓN DE CASOS DE USO DEL ACTOR:SECRETARIA

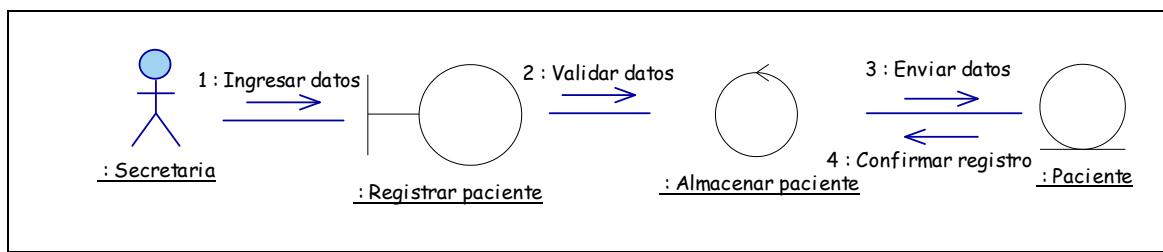
- **Diagrama de Colaboración del Cu: Autentificar secretaria**



Fuente: Propia

**Figura 3.54.** Diagrama de colaboración del CU: Autentificar secretaria

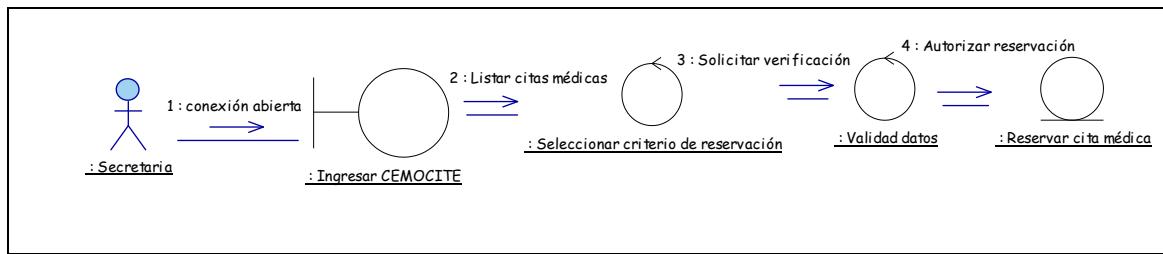
- **Diagrama de Colaboración del Cu: Ingresar Paciente**



Fuente: Propia

**Figura 3.55.** Diagrama de colaboración del CU: Ingresar paciente

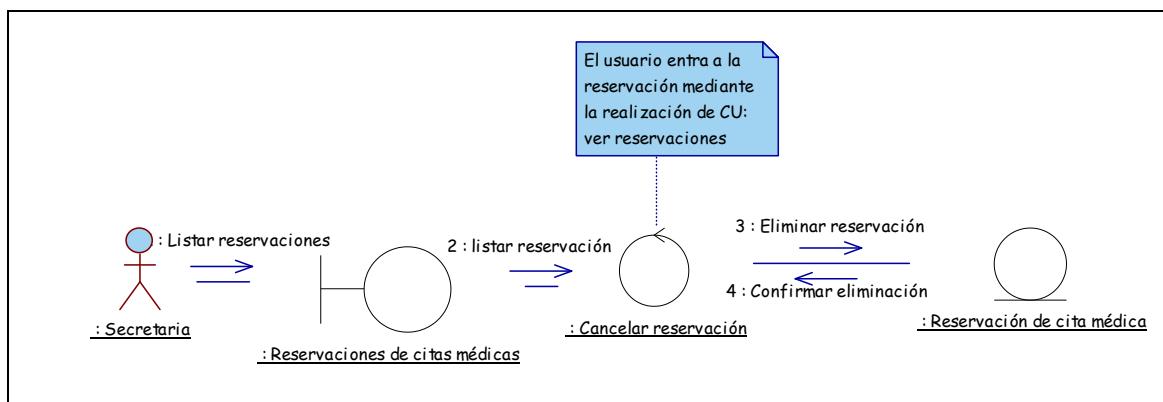
- **Diagrama de Colaboración del Cu: Reservar cita médica solicitada por el paciente**



Fuente: Propia

**Figura 3.56.** Diagrama de colaboración del CU: Reservar cita médica solicitada por el paciente

- **Diagrama de Colaboración del Cu: Cancelar reservación de cita médica**

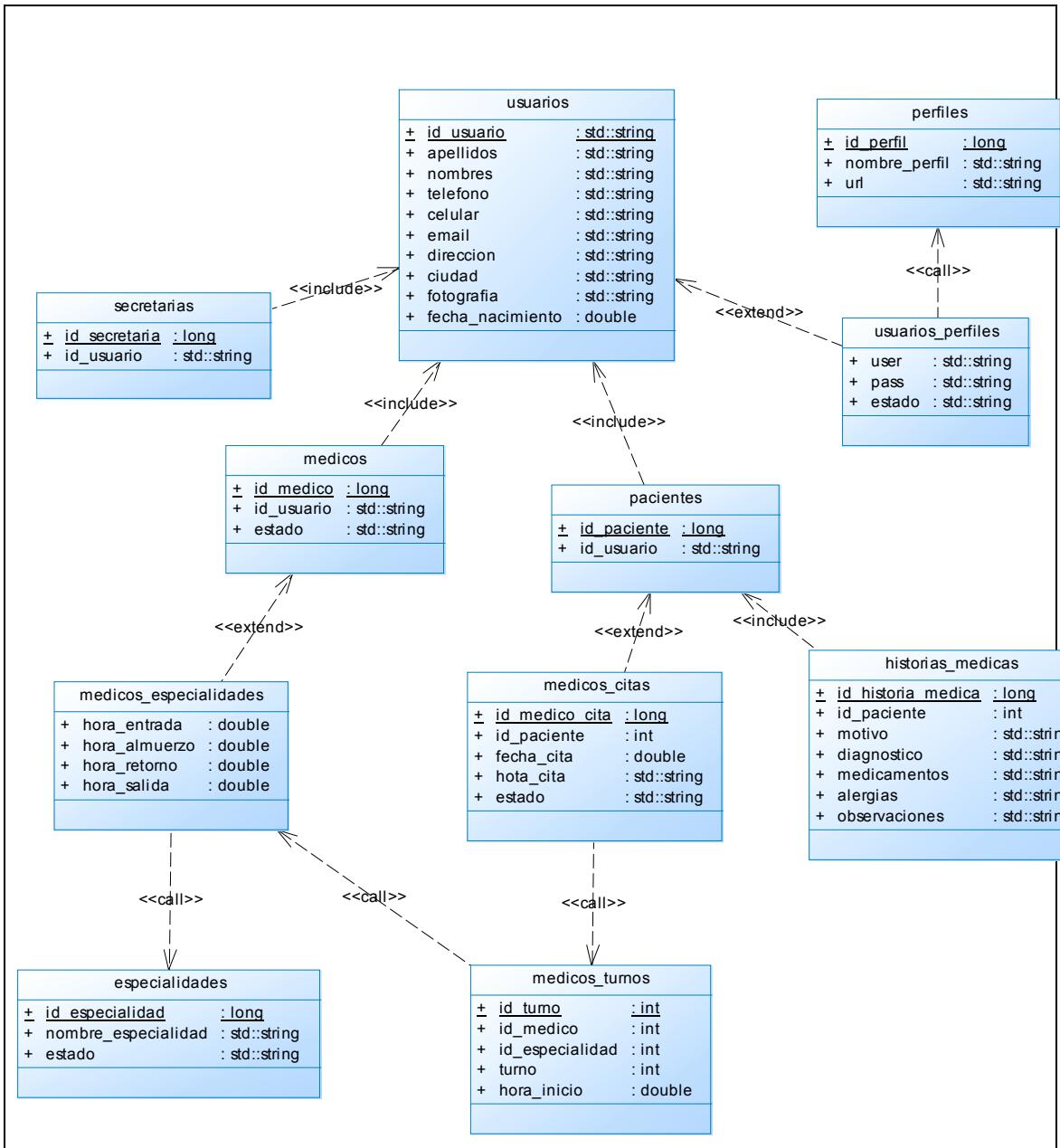


Fuente: Propia

**Figura 3.57.** Diagrama de colaboración del CU: Cancelar Reservación de cita médica

### 3.5.4 Diagrama de clases de diseño

En la figura se muestra el modelo total de las clases de diseño; la misma que incluye relaciones de dependencia y asociación, estas relaciones son las que probablemente conducirán al modelo entidad-relación de la base de datos.



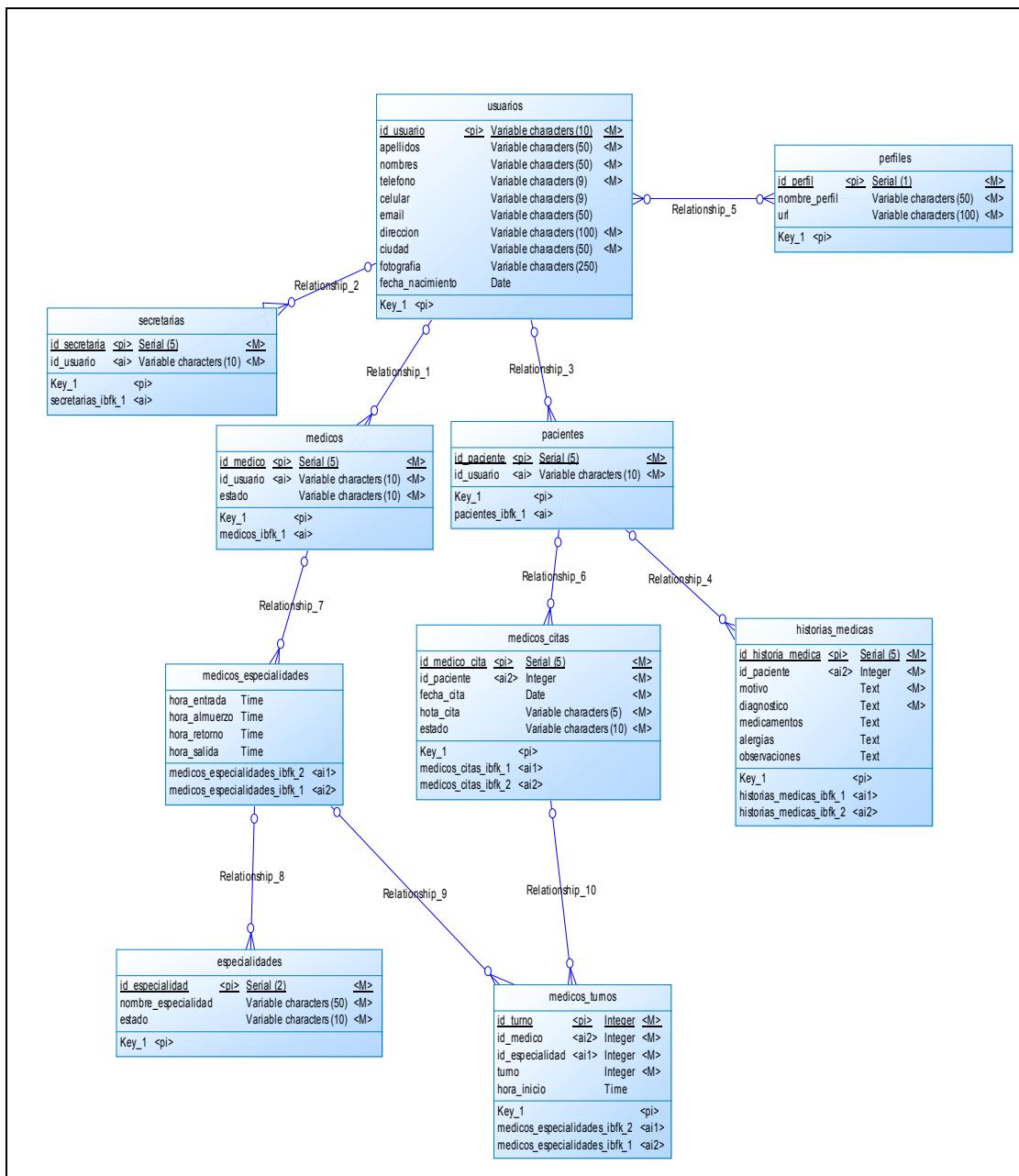
Fuente: Propia

**Figura 3.58.** Diagrama de clases del sistema CEMOCITE

### 3.5.5 Modelo de datos

Un modelo de datos es básicamente una descripción de algo conocido como contenedor de datos, así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas, son abstracciones que permiten la implementación de un sistema eficiente de base de datos, por ello presentamos el modelo físico y lógico de la base de datos del sistema CEMOCITE.

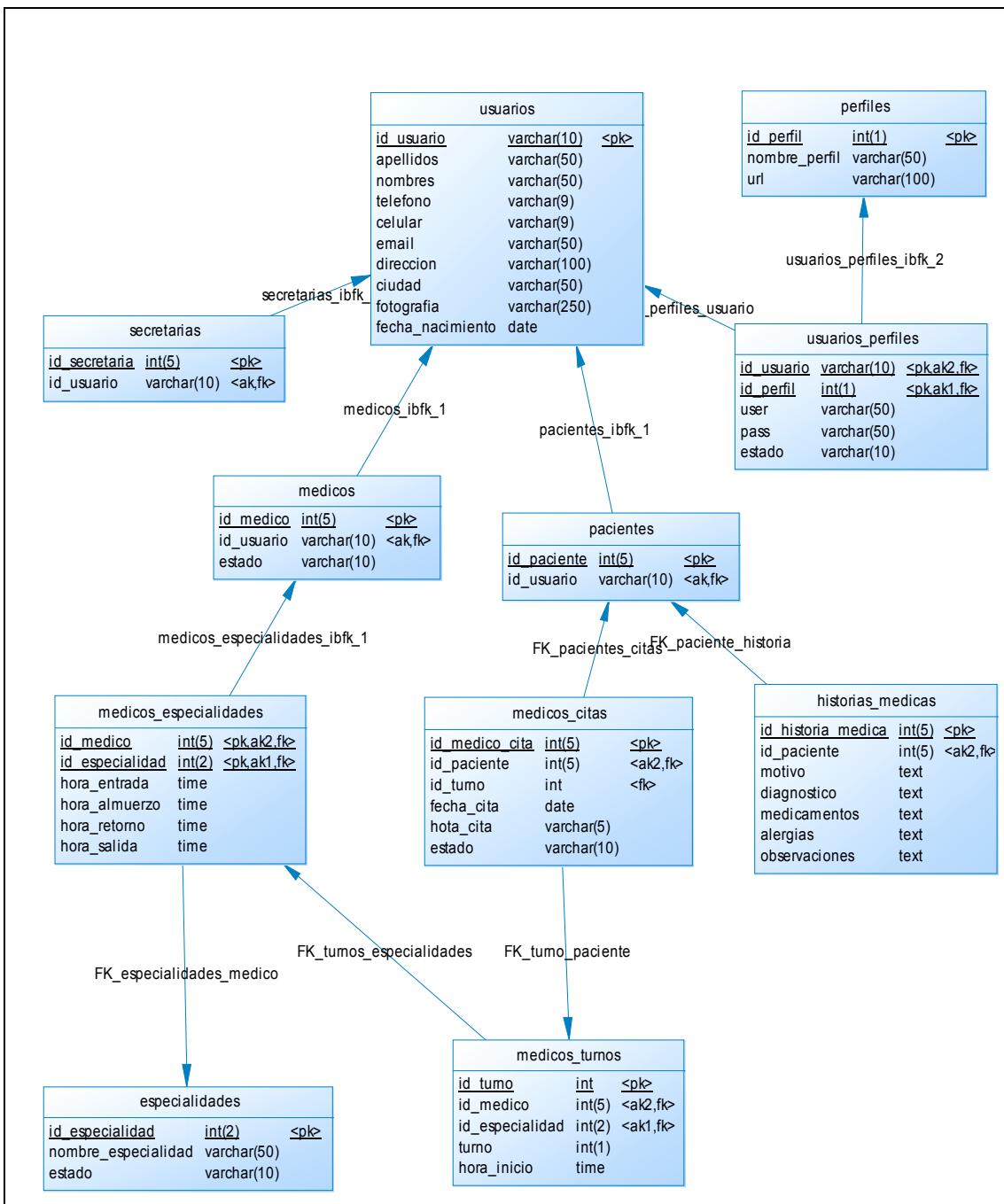
## Modelo Lógico de la Base de Datos



Fuente: Propia

**Figura 3.59.** Modelo lógico de la base de datos

## Modelo Físico de la Base de Datos



Fuente: Propia

**Figura 3.60.** Modelo físico de la base de datos

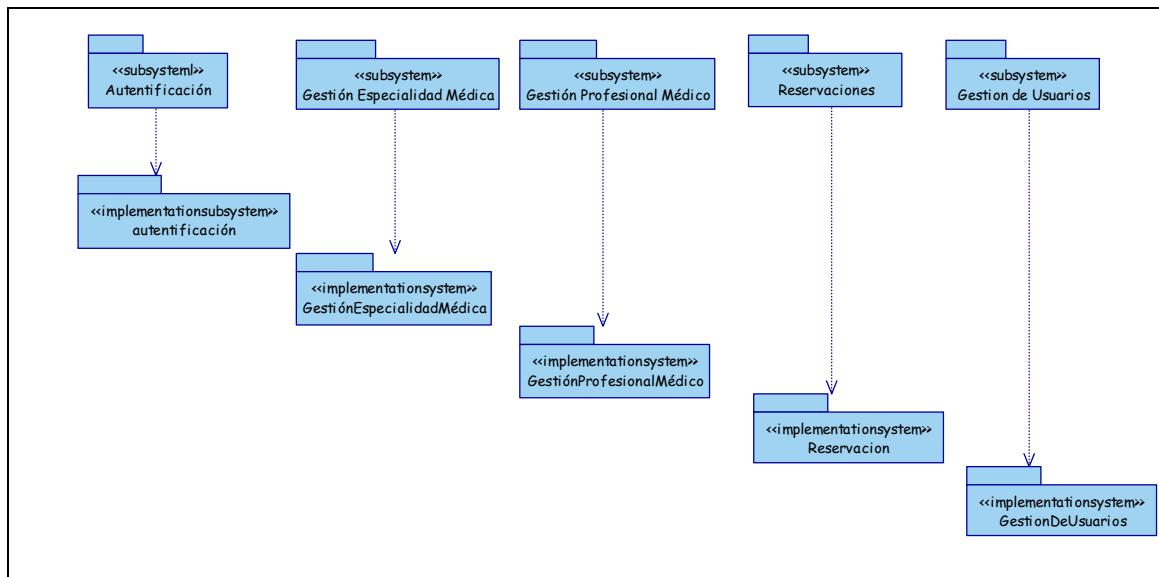
## 3.6 IMPLEMENTACION

La implementación inicia con el resultado del diseño. Este modelo es una colección de componentes y los subsistemas que los contienen, la implementación debe ser incremental, es decir en cada momento solo se añade un elemento, de este modo es más fácil localizar fallas y los componentes son probados más a fondo.

### 3.6.1 Definición de subsistemas de implementación

Los subsistemas de implementación se manifiestan a través de un mecanismo de empaquetamiento concreto en un entorno de implementación determinado.

Los subsistemas de implementación están relacionados con los subsistemas de diseño, formado por varios componentes que a su vez mantienen una relación directa con las clases que forman parte de los subsistemas.



Fuente: Propia

**Figura 3.61.** Modelo de Implementación

### 3.6.2 Desarrollo de la aplicación

Después de realizar el análisis y establecer el diseño del sistema se llegó a constituir cómo será el Sistema de Administración de Citas Médicas, de acuerdo a los requerimientos establecidos por todos los participantes tanto internos como externos.

En primera instancia se tiene la pantalla de inicio en la cual se ingresara el usuario y el password, para que así el sistema pueda verificar que tipo de perfil de usuario es y permitirle realizar las tareas asignadas para este usuario.



Fuente: Propia

**Figura 3.62.** Pantalla de ingreso al sistema

Cuando se ha verificado el usuario y el password se ingresara al sistema de administración de citas médicas. El contenido de cada una de las tareas del sistema lo desarrollamos en el Manual de Usuario que consta como (**Anexo 3**).

### 3.6.3 Implementación del sitio web

El Sistema de Administración de Citas Online se lo implementó en un Hosting en la Web el mismo que fue previamente contratado. El dominio bajo el cual se publica el sistema es [www.cemoplafotavaloo.org](http://www.cemoplafotavaloo.org), la implementación en el servidor web del hosting contratado se detalla paso a paso en el Manual de Instalación, que consta como (**Anexo 5**).

## 3.7 PRUEBAS AL SISTEMA

La prueba del software es un elemento crítico para la garantía de la calidad del software. El objetivo de la etapa de pruebas es verificar que el comportamiento externo del sistema CEMOCITE satisface los requisitos funcionales establecidos y especificados mediante los casos de uso. Además, esta etapa implica:

- Verificar la interacción de componentes.
- Verificar la integración adecuada de los componentes.
- Verificar que todos los requisitos se han implementado correctamente.

- Identificar y asegurar que los defectos encontrados se han corregido antes de entregar el software al cliente.

Diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

Para cumplir con estos objetivos se realizó un diseño de los casos de prueba que sirvió como guía para ejecutar las pruebas.

### **3.7.1 Casos de prueba**

Los casos de prueba permitirán probar tanto la funcionalidad del sistema como la integración de los componentes. En la siguiente tabla se especifica el formato para describir cada caso de uso de prueba.

No. Caso de Prueba	Número de Caso de Prueba
<b>Usuario</b>	Usuario ejecutor del caso de uso
<b>Nombre de Caso de Uso</b>	Nombre de Caso de Uso
<b>Nombre de Caso de Prueba</b>	Nombre de Caso de Prueba
<b>Entradas</b>	Datos que se ingresan al sistema, si aplica
<b>Salidas</b>	Datos que entregarán el sistema, si aplica
<b>Descripción</b>	Descripción breve del caso de prueba
<b>Procedimiento de Prueba</b>	Secuencia de pasos para realizar la prueba
<b>Resultados Esperados</b>	Descripción del comportamiento ideal del sistema durante la ejecución del procedimiento de prueba

Fuente: Propia

**Tabla 3.28.** Formato de descripción de casos de prueba

- **Prueba del caso de uso: Autenticar usuario administrador.**

No. Caso de Prueba	1
<b>Usuario</b>	Administrador
<b>Nombre de Caso de Uso</b>	Autenticar usuario Administrador
<b>Nombre de Caso de Prueba</b>	Autenticación del usuario administrador en el sistema
<b>Entradas</b>	Login: admin Clave: admin
<b>Salidas</b>	Bienvenida al sistema
<b>Descripción</b>	Ingreso al sistema administrador
<b>Procedimiento de Prueba</b>	<ol style="list-style-type: none"><li>1. Conectarse al sistema</li><li>2. Iniciar sesión</li><li>3. Ingrese login y password</li><li>4. Presionar botón iniciar sesión</li></ol>
<b>Resultados Esperados</b>	Ingresar al sistema administrador

Fuente: Propia

**Tabla 3.29.** Caso de prueba: Autenticar usuario administrador

- **Prueba del caso de uso: Cambiar password del administrador**

No. Caso de Prueba	2
Usuario	Administrador
Nombre de Caso de Uso	Cambiar password Administrador
Nombre de Caso de Prueba	Autentificación del usuario administrador en el sistema
Entradas	Login: admin Clave: admin Nueva Clave: administrador Confirmar Clave: administrador
Salidas	Bienvenida al sistema
Descripción	El administrador obtiene su nueva clave
Procedimiento de Prueba	1. Conectarse al sistema 2. Luego de haber ingresado como admin 3. Presione la opción cambiar clave 4. Ingrese clave actual, nueva clave y confirme.
Resultados Esperados	Cambio de clave e ingreso al sistema administrador

Fuente: Propia

**Tabla 3.30.** Caso de prueba: Cambiar password del administrador

- **Prueba del caso de uso: Registrar especialidad médica**

No. Caso de Prueba	3
Usuario	Administrador
Nombre de Caso de Uso	Registrar especialidad médica
Nombre de Caso de Prueba	Registro de una especialidad médica
Entradas	Nombre: Pediatría
Salidas	Visualización de especialidades
Descripción	El administrador ingresa una nueva especialidad
Procedimiento de Prueba	1. Ingresa al sistema como administrador 2. En el menú seleccionar especialidades 3. Presiona el botón nuevo 4. Ingresa la especialidad
Resultados Esperados	Ingreso de una nueva especialidad al sistema.

Fuente: Propia

**Tabla 3.31.** Caso de prueba: Registrar especialidad médica

- **Prueba del caso de uso: Listar especialidad médica**

No. Caso de Prueba	4
Usuario	Administrador
Nombre de Caso de Uso	Listar especialidad médica
Nombre de Caso de Prueba	Listar las especialidad médica
Entradas	Visualización de especialidades
Salidas	Lista de especialidades médicas
Descripción	El administrador ingresa una nueva especialidad
Procedimiento de Prueba	1. Ingresa al sistema como administrador 2. En el menú escoja especialidades 3. Lista las especialidades existentes
Resultados Esperados	Lista de especialidad al sistema.

Fuente: Propia

**Tabla 3.32.** Caso de prueba: Listar especialidad médica

- **Prueba del caso de uso: cambiar estado especialidad médica**

No. Caso de Prueba	5
Usuario	Administrador
Nombre de Caso de Uso	Cambiar estado especialidad médica
Nombre de Caso de Prueba	Modificación del estado de una especialidad médica
Entradas	Selección de estado
Salidas	Nombre de la especialidad, estado: activo o inactivo
Descripción	El administrador puede cambiar el estado de la especialidad
Procedimiento de Prueba	<ol style="list-style-type: none"> <li>1. Ingreso a especialidades</li> <li>2. Seleccionar en el combo box el estado</li> <li>3. Presiona el actualizar</li> </ol>
Resultados Esperados	Cambio del estado de la especialidad

Fuente: Propia

**Tabla 3.33.** Caso de prueba: Cambiar especialidad médica

- **Prueba del caso de uso: Ingreso de usuarios estándar**

No. Caso de Prueba	6
Usuario	Administrador
Nombre de Caso de Uso	Ingreso de usuarios estándar
Nombre de Caso de Prueba	Ingreso de médicos secretarias y pacientes
Entradas	Datos de los usuario
Salidas	Visualización de datos ingresados
Descripción	Ingreso de usuarios al sistema
Procedimiento de Prueba	<ol style="list-style-type: none"> <li>1. Ingreso a los menús respectivos</li> <li>2. Presiono el botón Agregar</li> <li>3. Presiono botón guardar</li> </ol>
Resultados Esperados	Ingreso de nuevos usuarios al sistema

Fuente: Propia

**Tabla 3.34.** Caso de prueba: Ingreso de usuarios estándar

- **Prueba del caso de uso: Autenticar usuario paciente**

No. Caso de Prueba	7
Usuario	Paciente
Nombre de Caso de Uso	Autentificar usuario Paciente
Nombre de Caso de Prueba	Autentificación del usuario paciente en el sistema
Entradas	Login: 1004442552 Clave: liss
Salidas	N/A
Descripción	Ingreso al sistema
Procedimiento de Prueba	<ol style="list-style-type: none"> <li>1. Conectarse al sistema</li> <li>2. Iniciar sesión</li> <li>3. Ingrese login y password</li> <li>4. presionar botón iniciar sesión</li> </ol>
Resultados Esperados	Ingresar al sistema

Fuente: Propia

**Tabla 3.35.** Caso de prueba: Autentificar usuario paciente

- Prueba del caso de uso: Reservar cita médica por especialidad y médico.**

No. Caso de Prueba	8
Usuario	Paciente
Nombre de Caso de Uso	Reservar cita médica por especialidad y médico
Nombre de Caso de Prueba	Reservar cita médica
Entradas	Selección de especialidad médica y de médico
Salidas	Horarios de atención
Descripción	Solicitud de cita médica
Procedimiento de Prueba	<ol style="list-style-type: none"> <li>1. Ingreso a solicitud de citas</li> <li>2. Selecciono la especialidad</li> <li>3. Selecciono el médico</li> <li>4. Escojo horario</li> <li>5. Confirmo cita</li> </ol>
Resultados Esperados	Reservo la cita y recibo correo de notificación de reservación

Fuente: Propia

**Tabla 3.36.** Caso de prueba: Reservar cita médica por especialidad y médico

- Prueba del caso de uso: Cancelar reservación por el usuario paciente**

No. Caso de Prueba	9
Usuario	Paciente
Nombre de Caso de Uso	Cancelar reservación por el usuario paciente
Nombre de Caso de Prueba	Cancelar reservación
Entradas	Ingreso al menú citas sin confirmar
Salidas	Despliega las citas sin confirmar
Descripción	Cancelo la cita
Procedimiento de Prueba	<ol style="list-style-type: none"> <li>1. Ingreso a citas sin confirmar</li> <li>2. Selecciono cancelar cita</li> <li>3. La cita se elimina</li> </ol>
Resultados Esperados	Eliminación de la cita médica

Fuente: Propia

**Tabla 3.37.** Caso de prueba: Cancelar reservación por el usuario paciente

- Prueba del caso de uso: Autenticar médico**

No. Caso de Prueba	10
Usuario	Médico
Nombre de Caso de Uso	Autentificar usuario Médico
Nombre de Caso de Prueba	Autentificación del usuario médico en el sistema
Entradas	Login: 1003451067 Clave: gomez
Salidas	N/A
Descripción	Ingreso al sistema
Procedimiento de Prueba	<ol style="list-style-type: none"> <li>1. Conectarse al sistema</li> <li>2. Iniciar sesión</li> <li>3. Ingrese login y password</li> <li>4. presionar botón iniciar sesión</li> </ol>
Resultados Esperados	Ingresa al sistema

Fuente: Propia

**Tabla 3.38.** Caso de prueba: Autentificar médico

- **Prueba del caso de uso: Ver reservaciones médicas asignadas al profesional médico**

No. Caso de Prueba	11
Usuario	Médico
Nombre de Caso de Uso	Ver reservaciones médicas asignadas al profesional médico
Nombre de Caso de Prueba	Ver citas del día
Entradas	Selección en el menú citas del día
Salidas	Lista de citas para el día
Descripción	Las reservaciones hechas por los pacientes
Procedimiento de Prueba	<ol style="list-style-type: none"> <li>1. Selecciono citas del día</li> <li>2. En acciones ingreso al historial del paciente</li> <li>3. Registro información y la guardo</li> <li>4. La cita fue concluida</li> </ol>
Resultados Esperados	Cita atendida por el médico

Fuente: Propia

**Tabla 3.39.** Caso de prueba: Ver reservaciones médicas asignadas al profesional médico

- **Prueba del caso de uso: Autenticar secretaria**

No. Caso de Prueba	12
Usuario	Secretaria
Nombre de Caso de Uso	Autentificar usuario Secretaria
Nombre de Caso de Prueba	Autentificación del usuario secretaria en el sistema
Entradas	Login: 1002951067 Clave: alejita
Salidas	N/A
Descripción	Ingreso al sistema
Procedimiento de Prueba	<ol style="list-style-type: none"> <li>1. Conectarse al sistema</li> <li>2. Iniciar sesión</li> <li>3.Ingrese login y password</li> <li>4. presionar botón iniciar sesión</li> </ol>
Resultados Esperados	Ingresa al sistema

Fuente: Propia

**Tabla 3.40.** Caso de prueba: Autentificar secretaría

- **Prueba del caso de uso: Ver reservaciones de citas médicas**

No. Caso de Prueba	13
Usuario	Secretaria
Nombre de Caso de Uso	Ver reservación de citas médicas
Nombre de Caso de Prueba	Solicitud de citas
Entradas	Selección en el menú
Salidas	Lista de citas por confirmar
Descripción	Confirmar la cita solicitada por el paciente
Procedimiento de Prueba	<ol style="list-style-type: none"> <li>1. Ingresar a solicitud de citas</li> <li>2. Confirmar cita</li> <li>3. El sistema envía notificación de reservación vía correo y sms.</li> </ol>
Resultados Esperados	Confirmación de cita.

Fuente: Propia

**Tabla 3.41.** Caso de prueba: Ver reservaciones de citas médicas

- Prueba del caso de uso: Reservar cita médica solicitada por el paciente**

No. Caso de Prueba	14
Usuario	Secretaria
Nombre de Caso de Uso	Reservar cita médica solicitada por el paciente
Nombre de Caso de Prueba	Secretaria reserva cita
Entradas	Cedula del paciente, selección especialidad y médico y luego la hora de la cita
Salidas	Cita solicitada
Descripción	Reservación de cita para un paciente
Procedimiento de Prueba	<ol style="list-style-type: none"> <li>1. Ingreso de cedula</li> <li>2. Selección de médico y especialidad</li> <li>3. Selección de hora de atención</li> <li>4. Sita solicitada</li> </ol>
Resultados Esperados	Solicitar cita médica

Fuente: Propia

**Tabla 3.42.** Caso de prueba: Reservar cita solicitada por el paciente

- Prueba del caso de uso: Eliminar reservaciones de citas médicas**

No. Caso de Prueba	15
Usuario	Secretaria
Nombre de Caso de Uso	Eliminar reservación de citas médicas
Nombre de Caso de Prueba	Eliminar cita
Entradas	Selección en el menú citas pendientes
Salidas	Lista de citas, eliminar cita
Descripción	Eliminar cita solicitada por el paciente
Procedimiento de Prueba	<ol style="list-style-type: none"> <li>1. Ingresar a solicitud de citas</li> <li>2. Eliminar cita</li> <li>3. El sistema envía notificación de eliminación vía correo y sms.</li> </ol>
Resultados Esperados	Confirmación de cita.

Fuente: Propia

**Tabla 3.43.** Caso de prueba: Eliminar reservaciones de citas médicas

### 3.7.2 Evaluación de los resultados de pruebas

Para la evaluación de resultados se ha considerado que cada tipo de usuario interactué por cada uno de los módulos del sistema CEMOCITE guiados por los casos de prueba, obteniendo una idea clara del resultado que se ha obtenido en las pruebas. En la siguiente tabla se describe el formato de presentación de evaluación de resultados.

Perfil de Usuario	Tipo de Usuario
Caso de Prueba	Número y Nombre del Caso de Prueba
Resultados	Descripción del comportamiento del usuario frente al sistema
Caso de Prueba	Número y Nombre del Caso de Prueba
Resultados	Descripción del comportamiento del usuario frente al sistema
Conclusión Final de la Evaluación	<b>MUY SATISFACTORIO:</b> usuario utiliza el sistema sin inconveniente y el sistema presenta un adecuado funcionamiento ante lo requerido  <b>SATISFACTORIO:</b> usuario utiliza el sistema sin mayores dificultades o dificultades que pueden ser y/o fueron solventadas mediante aprendizaje y/o el sistema presenta un funcionamiento adecuado con situaciones de menor complejidad y que pueden ser considerados requerimientos a posteriori  <b>INSATISFACTORIO:</b> Usuario presenta graves dificultades en el uso del sistema que no pueden ser solventadas mediante aprendizaje y/o el sistema no implementa la funcionalidad requerida por el usuario para cumplir sus tareas

Fuente: Propia

**Tabla 3.44.** Diseño de la tabla evaluación de resultados de prueba

#### **Tipo de usuario: Administrador**

Perfil de Usuario	Administrador
Caso de Prueba	Autentificar usuario administrador
Resultados	Muy Satisfactorio
Caso de Prueba	Cambiar password administrador
Resultados	Muy Satisfactorio
Caso de Prueba	Registrar especialidad médica
Resultados	Muy Satisfactorio
Caso de Prueba	Listar especialidad médica
Resultados	Muy Satisfactorio
Caso de Prueba	Cambiar estado de especialidad médica
Resultados	Muy Satisfactorio
Caso de Prueba	Ingreso de Usuarios estándar
Resultados	Muy Satisfactorio
Conclusión Final de la Evaluación	<b>MUY SATISFACTORIO</b>

Fuente: Propia

**Tabla 3.45.** Evaluación de resultados de pruebas para el usuario administrador

### Tipo de usuario: Usuario Paciente

Perfil de Usuario	Tipo de Usuario
Caso de Prueba	Autentificar usuario Paciente
Resultados	Muy Satisfactorio
Caso de Prueba	Reservar cita médica por especialidad y médico
Resultados	Muy Satisfactorio
Caso de Prueba	Cancelar reservación por el usuario final
Resultados	Muy Satisfactorio
Conclusión Final de la Evaluación	MUY SATISFACTORIO

Fuente: Propia

**Tabla 3.46.** Evaluación de resultados de pruebas para el paciente

### Tipo de usuario: Usuario Médico

Perfil de Usuario	Tipo de Usuario
Caso de Prueba	Autentificar médico
Resultados	Muy Satisfactorio
Caso de Prueba	Ver reservaciones de citas asignadas al médico
Resultados	Muy Satisfactorio
Conclusión Final de la Evaluación	MUY SATISFACTORIO

Fuente: Propia

**Tabla 3.47.** Evaluación de resultados de pruebas para el médico

### Tipo de usuario: Usuario Secretaria

Perfil de Usuario	Tipo de Usuario
Caso de Prueba	Autentificar secretaria
Resultados	Muy Satisfactorio
Caso de Prueba	Ver reservación de citas médicas
Resultados	Muy Satisfactorio
Caso de Prueba	Reservar cita médica solicitada por el paciente
Resultados	Muy Satisfactorio
Caso de Prueba	Eliminar reservaciones de citas médicas
Resultados	Muy Satisfactorio
Conclusión Final de la Evaluación	MUY SATISFACTORIO

Fuente: Propia

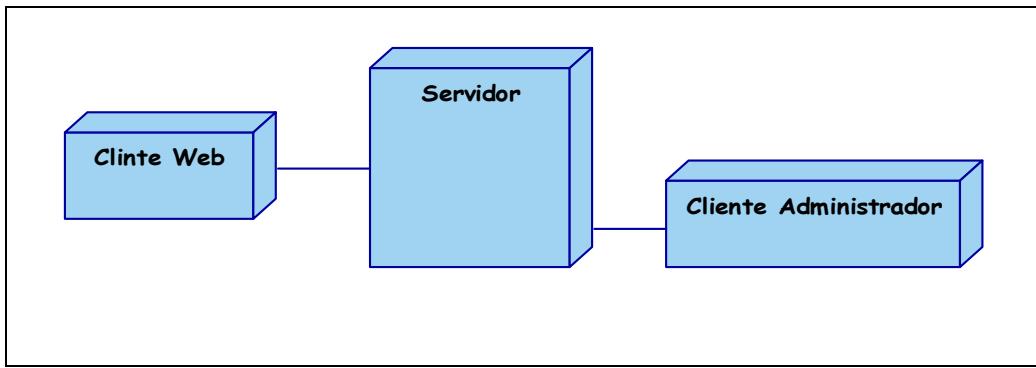
**Tabla 3.48.** Evaluación de resultados de pruebas para la secretaria

## 3.8 DESPLIEGUE

Este modelo muestra el despliegue la configuración de tipos de nodos del sistema, en los cuales se hará el despliegue de los componentes.

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos

de cómputo. Presenta los nodos interconectados y las clases activas que se ejecutan entre ellos, identificados durante el diseño. CEMOCITE se ejecutará en tres nodos como se muestra en la siguiente figura.



Fuente: Propia

**Figura 3.63.** Diagrama de Despliegue para el sistema CEMOCITE

### **3.8.1 Plan de Despliegue**

Este tiene como objetivo producir con éxito distribuciones del producto y distribuirlo a los usuarios. Las actividades implicadas incluyen:

- Probar el producto en su entorno de ejecución final, en nuestro caso subirlo a la web.
- Proveer asistencia y ayuda a los usuarios, esto se lo realiza con la capacitación a cada uno de los actores que intervienen en la utilización del sitio web.
- Formar a los usuarios y al cuerpo de publicidad, para dar a conocer el nuevo servicio que presta CEMOPLAF Otavalo.
- Migrar la información existente o convertir a bases de datos, esto se realiza para la migración de datos existentes en CEMOPLAF donde se encuentra la información concerniente a los usuarios.

Una vez que el producto de software ha sido implementado y probado exitosamente, es momento de llevar el producto al cliente. El propósito es producir releases del producto y llevar el software a los usuarios finales.

### **3.8.2 Paso a producción**

El presente procedimiento tiene como objetivo establecer y describir las actividades a ejecutar a fin de autorizar el pase a producción del Sistema Dewsarrollado.

A fin de establecer el pase a producción es necesario que se cumplan las siguientes actividades:

a) **Revisión de documentación y entrega de Programas Fuentes:** Revisar que la documentación y programas fuentes entregados sean consistentes.

Documentos:

- Acta de Aprobación Usuarios
- Guía de referencia de instalación
- Manual de Usuario
- Manual Técnico del Sistemas
- Acta(s) capacitación usuarios y sus nombres
- Documento de creación de usuarios y privilegios

b) **Desarrollo Informe de Pase a producción:** Se deberá generar el informe correspondiente y gestionar las autorizaciones respectivas.

Puntos:

- Check list del cumplimiento de la documentación y programas fuentes
- Determinación de los servidores a utilizar
- Estimación de cálculo de espacio requerido
- Requerimientos técnicos exigidos: hardware, software y configuración.

En la siguiente tabla se detalla la aprobación para el paso a producción, una vez que se entregó todos los documentos necesarios para el usuario y la certificación de las pruebas realizadas por cada uno de los usuarios definidos en el proyecto.

Aprobado por:	Dra. LILIAN CERVANTES												
Tema:	CEMOCITE												
Detalle:	Sistema Integrado de Reservación y Notificación de Citas Médicas Online												
Revisión de documentación y entrega de Programas Fuentes	<table><tr><td>Acta de Aprobación Usuarios</td><td>Aprobado</td></tr><tr><td>Guía de referencia de instalación</td><td>Ok</td></tr><tr><td>Manual de Usuario</td><td>Ok</td></tr><tr><td>Manual Técnico del Sistemas</td><td>Ok</td></tr><tr><td>Acta(s) capacitación usuarios y sus nombres</td><td>Ok</td></tr><tr><td>Documento de creación de usuarios y privilegios</td><td>Ok</td></tr></table>	Acta de Aprobación Usuarios	Aprobado	Guía de referencia de instalación	Ok	Manual de Usuario	Ok	Manual Técnico del Sistemas	Ok	Acta(s) capacitación usuarios y sus nombres	Ok	Documento de creación de usuarios y privilegios	Ok
Acta de Aprobación Usuarios	Aprobado												
Guía de referencia de instalación	Ok												
Manual de Usuario	Ok												
Manual Técnico del Sistemas	Ok												
Acta(s) capacitación usuarios y sus nombres	Ok												
Documento de creación de usuarios y privilegios	Ok												
Desarrollo Informe de Pase a producción	<table><tr><td>Cumplimiento de la documentación</td><td>Aprobado</td></tr><tr><td>Determinación de los servidores a utilizar</td><td>Ok</td></tr><tr><td>Estimación de cálculo de espacio requerido</td><td>Ok</td></tr><tr><td>Requerimientos técnicos: hardware, software y configuración.</td><td>Ok</td></tr></table>	Cumplimiento de la documentación	Aprobado	Determinación de los servidores a utilizar	Ok	Estimación de cálculo de espacio requerido	Ok	Requerimientos técnicos: hardware, software y configuración.	Ok				
Cumplimiento de la documentación	Aprobado												
Determinación de los servidores a utilizar	Ok												
Estimación de cálculo de espacio requerido	Ok												
Requerimientos técnicos: hardware, software y configuración.	Ok												

Fuente: Propia

**Tabla 3.49. Actividades de paso a producción**

### **3.8.3 Transferencia de tecnología y capacitación**

La transferencia de tecnología se la realiza con cada uno de los actores involucrados en el sitio web, además se proveerá de los documentos necesarios para el buen manejo de la aplicación como son:

- Manual de Usuario
- Manual Técnico Funcional
- Manual de Instalación

Los manuales están incluidos en los anexos: **(Anexo3), (Anexo4), (Anexo5)**,

# CAPÍTULO 4

## CONCLUSIONES Y RECOMENDACIONES



*Conclusiones*

*Recomendaciones*

## **4.1 CONCLUSIONES**

- Cemoplaf Otavalo cuenta con un gran número de pacientes, al brindarles una aplicación web para la administración online de citas médicas, descongestionará los pasillos y brindará un mejor servicio a los usuarios.
- La Aplicación web para la administración online de citas médicas tiene que brindar de forma rápida, transparente y segura la administración de citas médicas al gestionar correctamente las diferentes cuentas de usuario según los perfiles establecidos incrementando la seguridad e integridad de la información.
- RUP, como la metodología más adaptable a los requerimientos en el desarrollo del proyecto; proporcionó un acercamiento disciplinado a la asignación de tareas y responsabilidades al realizar el sistema de reservación de citas médicas; asignando roles a cada actor, administrador, médicos, pacientes o usuarios del sistema y estableciendo algunos de los documentos (Visión); necesarios para el establecimiento de los parámetros como el alcance, requerimientos y especificaciones básicas del sistema; que se ajuste a las necesidades de los usuarios finales con unos costo y un calendario predecible; utiliza los Casos de Uso para guiar el proceso, se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.
- Las necesidades del centro médico y de sus involucrados tanto internos como externos, se establecieron y definieron de una manera adecuada, tomando como base la información proporcionada por el centro médico y adaptándola a un modelo de negocio que pueda ser implementado en un sistema (Diagrama del Modelo de Negocio); para lo cual se hizo uso de varias alternativas existentes en el desarrollo de software a medida como lo son los diagramas UML dotando de una herramienta visual para la codificación del sistema; en este sentido la utilización de una herramienta que permita representar de una manera gráfica toda la lógica de la aplicación (StarUML) fue de mucha ayuda; a pesar de no contar con suficiente documentación de la misma en español; la forma de modelar y el entorno de la aplicación permitió la creación de la totalidad del modelado de la aplicación; Casos de uso, colaboración y secuencia.
- En nuestra aplicación utilizamos MySQL como base de datos ya que a pesar que PHP tiene como característica principal su soporte para una gran cantidad de bases de datos; MySQL es la que por mucho tiempo mejor se ha acoplado; tomando en cuenta que el centro médico no cuenta con un servidor que

disponga de una IP Pública es necesario para que el sistema esté disponible en Internet subirlo a un hosting; tomando como punto de referencia los hostings disponibles quen en gran mayoría cuentan con un Gestor de Base de Datos MySQL.

- La utilización de CodeIgniter como framework que permita la implementación de la aplicación, por las prestaciones que esto implica y bajo la arquitectura MVC en la que está desarrollado; permitió que la aplicación en su totalidad cumplir las necesidades descritas en el proyecto; y poder contar con una aplicación que por su separación de las capas pueda ser fácilmente mantenido o modificado; sin embargo al tratarse de un framework nuevo y muy poco conocido, en ciertos momentos fue complicado encontrar documentación como ejemplos y tutoriales dentro de los helpers, librerías y demás componentes de la aplicación; que fueron superado por la necesidad de investigar y presentar una alternativa en el desarrollo de aplicaciones Web bajo lenguaje PHP.
- Con el desarrollo de este trabajo de tesis se obtuvo un producto innovador, capaz de permitir que los pacientes en una institución médica tenga la facilidad de realizar la reservación a una cita en cualquier momento del día y en cualquier lugar; gestionándose mediante un sistema administrador que asegura la integridad y consistencia de los datos ingresados; además de presentar los horarios de atención disponibles de los profesionales médicos, evitando acudir físicamente y en muchos de los casos en vano a la institución médica, con tan solo acceder al sistema CEMOCITE se brinda una solución rápida y eficaz.
- El uso de software libre se dió por los beneficios que conlleva, frente a alternativas privadas, pudiéndose obtener incluso de forma gratuita y generar mediante mejoras soluciones altamente compatibles que cubren los requerimientos del sistema; son características de las herramientas utilizadas en gran parte de este trabajo tales como StarUML, CodeIgniter, MySQL, PHP, que sirvieron en el desarrollo de la aplicación y sujetándose a la metodología planteada en todas las fases del proyecto: Análisis, Diseño y Desarrollo de la misma.
- Durante el desarrollo del proyecto se pudo apreciar que las necesidades médicas van más allá del alcance del presente proyecto, permitiendo así la apertura de nuevos proyectos que complementen al sistema CEMOCITE, como por ejemplo la integración de todos los centro médicos a nivel provincial, regional y nacional para consolidar la información.

- Al automatizar los procesos que antes se realizaban manualmente se obtuvo un ahorro de tiempo y de costos al momento de realizar las transacciones.
- Los resultados finales obtenidos en la etapa de pruebas fueron satisfactorios, por lo tanto, se concluye que el sistema puede ser expuesto a nuevas implementaciones que permitan mejorar aún más el objetivo inicial del presente proyecto.

## **4.2 RECOMENDACIONES**

- Al desarrollar cualquier proyecto informático es necesario empaparse bien con toda la información posible del tema al cual se va a hacer referencia en el desarrollo de la aplicación de software, para ello sería necesario contar con las bases principales en el levantamiento de la información; y lograr un claro conocimiento de las herramientas con las cuales se desarrollará el proyecto, esto puede ser parte dentro del pensum de estudio dentro de la carrera universitaria, ya que al contar con mayor práctica se puede lograr un mejor adiestramiento en el desarrollo de software; y así ahorrar tiempo en cuanto al aprendizaje de herramientas que faciliten el proceso de elaboración de los sistemas logrando acortar los tiempos de entrega.
- Se tiene que capacitar a las personas que vayan a usar la Aplicación web para la administración online de citas médicas ya que deben tener conocimiento de las funcionalidades de la aplicación para poder aprovecharlas al máximo, y obtengan los beneficios previstos en el alcance y delimitación del proyecto.
- En el desarrollo de futuros proyectos de desarrollo de software se debería ahondar más exhaustivamente en la investigación de herramientas de software libre por los beneficios que éstas nos brindan.
- Investigar otras implementaciones similares tanto en el aspecto de frameworks de desarrollo de aplicaciones, así como también en tecnologías relativamente nuevas acoplando ciertas soluciones tecnológicas en el desarrollo de nuevos productos a medida; esto puede ser temas para futuros proyectos de Tesis ya que los constantes cambios permiten disponer de mucho material a ser analizado.
- Dentro de los posibles temas relacionados al presente proyecto pueden ser: Consolidación de la información médica de los pacientes en una Base de Datos única manejada por todos los centros médicos Cemoplaf a nivel Provincial, Regional o Nacional; evitando la duplicación de la información y manteniendo un historial médico único para cada paciente. Otro tema que

puede desprenderse del presente proyecto va relacionado a las enfermedades mas comunes tratadas por el centro médico, que permita establecer una tendencia de sintomas y enfermedades frecuentes.

# GLOSARIO DE TÉRMINOS Y BIBLIOGRAFÍA



*Glosario de Términos*

*Bibliografía*

# GLOSARIO DE TÉRMINOS

**AJAX:** acrónimo de Asynchronous JavaScript And XML, es una técnica de desarrollo web para crear aplicaciones interactivas.

**Ansi-Sparc:** Siglas en inglés (American National Standard Institute - Standards Planning and Requirements Committee), Planificación Normas y Requisitos Comité, es un estándar de diseño abstracto para un Sistema de Gestión de Base de Datos (DBMS) propuesto por primera vez en 1975. Más moderna comercial DBMS se basan en este sistema. The ANSI-SPARC model however never became a formal standard. El modelo ANSI-SPARC sin embargo, nunca se convirtió en un estándar formal.

**APROFE:** Asociación Pro Bienestar de la Familia Ecuatoriana.

**Apache/BSD:** El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc).

**Barry Boehm:** Desarrollo el modelo en espiral, es un modelo de ciclo de vida del software definido por primera vez en 1986.

**CEO:** Siglas en Inglés (Chief Executive Officer) Oficial Ejecutivo en Jefe.

**Codelgniter:** es un Entorno de Trabajo para Aplicaciones.

**ExpressionEngine:** Es un producto desarrollado por la empresa Ellislab, ellos son los desarrolladores y propietarios del código, los responsables de mantenerlo y cuidarlo. Ellislab también ha creado Codelgniter y ofrece un servicio de hosting a través de EngineHosting.

**Ficheros HTML:** Un fichero es el soporte en el que se encuentran almacenados o registrados los datos de lenguaje de marcado de hipertexto.

**Helpers:** En Codelgniter, es una colección de funciones en una categoría en particular.

**HTML:** Siglas en Inglés (HyperText Markup) Lenguaje de formato de Documentos para Hipertexto.

**HTTP:** Siglas en Inglés ( HyperText Transfer Protocol), Protocolo de transferencia de Hipertexto.

**Initial Operational Capability:** Traducción en español, capacidad operacional inicial.

**IMAP:** Siglas en Inglés (Internet Message Access Protocol) Protocolo de Acceso a Mensajes Internet.

**JavaScript:** Es un lenguaje de programación interpretado o sea, no requiere compilación.

**Lifecycle Architecture:** Traducción en español, Ciclo de vida de la Arquitectura.

**Lifecycle Objectives:** Traducción en español, Ciclo de vida de los Objetivos.

**MySQL:** Es un acrónimo de open source database software y pertenece a la categoría Bases de Datos.

**NNTP:** Siglas en Inglés (Network News Transfer Protocol) Protocolo de Transferencia de Noticias en Redes.

**Open Source:** Código abierto es el término con el que se conoce al software distribuido y desarrollado libremente.

**POP3:** Siglas en Inglés (Post Office Protocol version 3) Protocolo de oficina de correos versión 3.

**Product Release:** Traducción en español, terminar o sacar un producto.

**Protocolos:** Es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red por medio de intercambio de mensajes.

**RUP:** Acrónimo en inglés (Rational Unified Process), es un proceso de desarrollo de software.

**SGBD:** Siglas en Inglés (database management system, abreviado DBMS) sistemas de gestión de bases de datos.

**RSS:** Siglas en inglés (Rich Site Summary) Sumario de Sitio Enriquecido.

**SNMP:** Siglas en Inglés (Simple Network Management Protocol) Protocolo para Administración Simple de Redes.

**sourceforge.net:** Página Web para publicar Código o Aplicaciones Open Source.

**UML:** Acrónimo en Inglés (Unified Modeling Language) Lenguaje Unificado de Modelado es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

**URI:** Siglas en Inglés (Universal Resource Identifier ), Identificador universal de recursos.

**URLs:** Universal Resource Locator Localizador Universal de Recursos.

**Versión beta:** Representa generalmente la primera versión completa del programa informático o de otro producto.

# BIBLIOGRAFÍA

- LIB01.* (2000). Kruchten, P.,The Rational Unified Proces: An Introduction. *Addison Wesley*.
- LIB02.* (1995). Kruchten,P.,Architchitecture Blueprints: The "4+1" View Model of Software Architecture. IEEE Software, pp. 42-50. *Addison Wesley*.
- LIB03.* (2000). Jacoboson, I.,Booch,G.,Rumbaugh, J., El Proceso Unificado de Desarrollo de Software. *Addison Wesley*.
- LIB04.* (1998). Rational Software Corporation, Rational Unified Process, Best Practices for Software Development Teams. *Addison Wrsley*.
- LIB05.* (2002). Rational Software Corporation, Product: Rational Software Corporation.
- LIB06.* (2004). UML con RATIONAL ROSE. In G. ROMERO MORENO, UML con RATIONAL ROSE. *Grupo Editorial Megabyte. Primera edición*.
- REV01.* (Enero 2012). *Revista Cemoplauf Otavalo.* 6.
- www01. (2005, Octubre 31). <http://www.eluniverso.com/>. Retrieved Abril 15, 2012, from <http://www.eluniverso.com/2005/10/31/0001/18/3513E48CA554468FAA3B015C2ACC7883.html>
- www02. (n.d.). <http://www.dsi.uclm.es/>. Retrieved Abril 15, 2012, from <http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-Guia%20RUP.pdf>
- www03. (n.d.). <http://es.wikipedia.org/>. Retrieved Abril 11, 2012, from [http://es.wikipedia.org/wiki/Proceso\\_Unificado\\_de\\_Rational](http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational)
- www04. (n.d.). <http://bibdigital.epn.edu.ec/>. Retrieved Abril 15, 2012, from <http://bibdigital.epn.edu.ec/bitstream/15000/263/1/CD-0685.pdf>
- www05. (n.d.). <https://pid.dsi.upv.es/>. Retrieved Abril 10, 2012, from <https://pid.dsi.upv.es/>
- www06. (1995, enero). <http://www.eici.ucm.cl/>. Retrieved Abril 15, 2012, from [http://www.eici.ucm.cl/Academicos/R\\_Villarroel/descargas/ing\\_sw\\_1/Cap1Sommerville.pdf](http://www.eici.ucm.cl/Academicos/R_Villarroel/descargas/ing_sw_1/Cap1Sommerville.pdf)
- www07. (n.d.). <http://www.librosweb.es/>. Retrieved Abril 12, 2012, from [http://www.librosweb.es/jobeet/capitulo4/la\\_arquitectura\\_mvc.html](http://www.librosweb.es/jobeet/capitulo4/la_arquitectura_mvc.html)
- www08. (2005). <http://sisinfo.itc.mx>. Retrieved Agosto 02, 2012, from [http://sisinfo.itc.mx/ITC-APIRGG/Fundamentos\\_PHP/Introduccion.htm](http://sisinfo.itc.mx/ITC-APIRGG/Fundamentos_PHP/Introduccion.htm)
- www09. (2011, Diciembre 01). <http://desarrollomaya.blogspot.com>. Retrieved Agosto 06, 2012, from <http://desarrollomaya.blogspot.com/2011/12/experiencia-con-php.html>
- www10. (2010). <http://www.ciberaula.com>. Retrieved Septiembre 05, 2012, from [http://www.ciberaula.com/curso/Cursos\\_de\\_PHP\\_On-Line/Ciberaula.mht](http://www.ciberaula.com/curso/Cursos_de_PHP_On-Line/Ciberaula.mht)
- www11. (2009, Febrero 16). <http://tednologia.com>. Retrieved Agosto 22, 2012, from <http://tednologia.com/mvc-en-php/>
- www12. (2012, Julio 01). <http://ebookbrowse.com>. Retrieved Agosto 03, 2012, from <http://ebookbrowse.com/codeigniter-guia-usuario-2-1-0-pdf-d310616078>
- www13. (2000, Noviembre 22). <http://www.bosque.udec.cl>. Retrieved Agosto 08, 2012, from <http://www.bosque.udec.cl/~sram/manuals/myphptut.pdf>

- www14. (2012, Septiembre 03). <http://es.wikipedia.org>. Retrieved Septiembre 09, 2012, from [http://es.wikipedia.org/wiki/Sistema\\_de\\_gesti%C3%B3n\\_de\\_bases\\_de\\_datos](http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_bases_de_datos)
- www15. (2012, Noviembre 28). <http://www.eclipsoft.com>. Retrieved Noviembre 30, 2012, from Eclipsoft: <http://www.eclipsoft.com/index.php/servicios-services/aplicaciones-moviles>

## **ANEXOS**



## **ANEXOS**

### **ANEXO 1**

“Copia del Anteproyecto de tesis por el consejo directivo de la facultad”

### **ANEXO 2**

“Detalle del documento de vision.doc”

### **ANEXO 3**

“Manual de Usuario.doc”

### **ANEXO 4**

“Manual Técnico Funcional.doc”

### **ANEXO 5**

“Manual de Instalación”